Development and maintenance of IT-systems in Norwegian organizations

Master Thesis,
Institute Of Informatics,
University of Oslo

Written by: Arthur Jahr (arthurj@ifi.uio.no)

Abstract

For a long time IT-system maintenance has been reckoned as the largest expense attached to the IT-department, but there has been relatively little research on this subject.

This investigation looks at IT-system maintenance by presenting results from a questionnaire performed in Norwegian organizations. We have looked at how the different organizations perform development and maintenance of their IT-systems. We hope the results can increase the knowledge within this area and improve the organizations ability to perform maintenance in a cost-effective way.

This investigation is the third in a series of such investigations, and that is why we compare our results with [Holgeid 99] and [Krogstie 94].

The share of maintenance-work is significantly larger than the share development-work. The share of maintenance compared to the share development is lower than in [Holgeid 99], but still relatively large compared to earlier investigations. In our investigation the amount of traditional maintenance is almost 66%, when we compare share traditional maintenance to share development.

Work related to development of functionality in IT-systems comes to 39%, and maintenance of the existing functionalities make up the remaining 61%. This is similar to the results in [Holgeid 99], but a big change compared to [Krogstie 94] where functional development was at 56%, and functional maintenance 44%. In our investigation, as in [Holgeid 99], functional maintenance is higher then functional development, while in [Krogstie 94] the opposite was reported.

The share of maintenance looks to have increased compared to earlier investigations, except [Holgeid 99]. Several conditions seem to influence share maintenance, one of them being the complexity of the portfolio. Organizations with a complex portfolio seem to have significantly less maintenance than organizations with less complex portfolios. Intuitively this may look unnatural, but similar features have been reported in earlier investigations (e.g. [Holgeid 99]).

Organizations which do not use pre-defined methods in their development and maintenance have more maintenance than organizations that use pre-defined methods.

Preface

The work on this investigation started in November 2002. After a slow start, the work picked up and the questionnaires were sent out in November 2003.

There has been a series of people who have contributed to this investigation. First I want to thank my advisors John Krogstie, SINTEF, and Dag Sjøberg, Institute Of Informatics. They have given me advice and support to help me with the work. Their knowledge and expertise on the subject has been of great importance to investigation.

Thanks to SIMULA for the financial support and the help with the practical work in connection to carrying out the investigation. I will also thank The Norwegian Computer Society, for giving me access to their member-list, from where the respondent organizations were picked out.

I would like to thank all that have participated in this investigation.

The University of Oslo, February 2005

Arthur Jahr

Table of contents

1 Introduction

Over the last 40-50 years there has been written an incredible amount of programming-code. The reason for this is that both the use and dependence of software has increased dramatically since more and more business sectors have started to use information-technology.

Large amounts of applications are developed, and not many of them were expected to be used and maintained for a quarter of a century, therefore were many of them not designed for long-lasting maintenance. This is one of the main reasons that maintenance of IT-systems has for many years been estimated as the biggest expense for the IT-department. Despite this, there has been very little research on this subject.

The increasing importance of IT in the society, and in different organizations, makes it important to do research on maintenance of IT-systems.

In this investigation we will present the results from a questionnaire about development and maintenance of IT-systems in Norwegian organizations. The questionnaire was made in the winter 2003/2004. Different condition around development and maintenance is examined, with emphasis on maintenance of IT-systems.

There are relative few investigations on this field, so we hope the results from our investigation can help the organizations that participated. Our investigation is especially interesting since it is possible to compare our results with the ones in investigations from 1993 [Krogstie] and 1998 [Holgeid 99]. Both of these investigations are similar to ours, with very similar respondent-group from The Norwegian Computer Society and comparable variables. The same survey form has been used in the three investigations, with some minor differences.

In connection to the investigation an article has been written - A Longitudinal Study of General and Functional Maintenance in Norway [Krogstie, Sjøberg, Jahr 05] – as an overview of this investigation.

1.1 Approach to the problem

With the large maintenance-costs, it is important to obtain more knowledge of how the maintenance-work is distributed on different types of maintenance tasks, and what factors affect the distribution.

In this investigation we will look at different condition around maintenance of IT-systems. We will look at how maintenance is affected by different conditions such as:

- Size and type of organization
- Importance of IT for the organization
- Internal qualifications in the IT-department
- Complexity of the application portfolio

- Organization of the system-developers
- Use of methods
- Use of tools

1.2 Research method

[Hale et al 91] reported that almost half of all research within system-development is student-research performed on other students. This is an easy way to conduct an investigation, but rarely transferable to real life. Our investigation is performed amongst real organizations and we hope to obtain a more valid result.

The data is collected with questionnaires that were distributed to organizational members of The Norwegian Computer Society.

The collected data has been used in different statistical analyses to test 24 hypotheses. We have looked at the results in context, and tried to find explanations for the results from the statistical analyses. We have also compared the results to earlier investigations to check for continuity and similarities.

1.3 Contribution

The results from the questionnaire give us an idée of how maintenance is performed in Norwegian organizations, and how the situation is today compared to earlier.

Some of our results are surprising, while other results are verification of earlier investigations. Both types of results are interesting when we try to increase our knowledge of how maintenance of IT-systems is performed today.

1.4 Outline of the report

In chapter 2 we look at the definitions of IT-system maintenance, some earlier investigations and life-cycles models and maintenance.

In chapter 3 we present the research methods used in the investigation, and we describe the different statistical techniques used.

Chapter 4 present the hypotheses that are tested in the investigation.

In chapter 5 we will show the descriptive results from the investigation. All the results will give us a good understanding about the situation in the participating organizations, which will help us in the testing of the hypotheses.

In chapter 6 we will present the results from the hypotheses-testing, compare them with results from earlier investigations, and discuss different aspects of the hypotheses-testing.

Chapter 7 is used to present a discussion of the main results, and we mention different explanation-models.

In chapter 8 we have an evaluation of the research method, and in chapter 9 we have the conclusion and further work.

2 Maintenance of IT-systems

In this chapter we will define different maintenance-concepts, and we will discuss different conditions attached to maintenance of IT-systems.

2.1 Definitions

There are many different definitions of IT-system maintenance. In the list under we will show some of these:

”Maintenance is fixing software bugs” [Reutter 81] ”Maintenance is the process of modifying existing operational software while leaving its summary functions intact” [Boehm 76] ”Maintenance is the mechanism for combating software deterioration, which over time tends to become unstructured, unreliable, and resistant to change” [Lyons 81] ”Maintenance is the activity associated with keeping operational computer system continuously in tune with the requirements of users, data processing operations, associated clerical functions and external demands from governmental and other agencies” [Riggs 69] ”Maintenance refers to modifying a program – updating an existing program’s functions to reflect new constraints or additional features” [Liu 76] ”Maintenance is adapting software to meet constantly changing business needs” [Bush 88] ”Maintenance includes updating as well as fixing bugs in existing applications” [Moad 90] ”Maintenance is the continuing process of keeping the program running, or improving its characteristics... Program modification has as its objective the adaption to a changing environment” [Ogden 72] ”Maintenance is the process of modifying a software system or component after delivery to correct faults, improve performance, or other attributes, or adapt to a changed environment” [IEEE 91] ”Maintenance consists of changes that need to be made to a computer program after the software been turned over to the customer or goes into production” [Scott and Farley 88] ”Maintenance is performed in response to system failures, to changes in data and processing requirements, to eliminate processing inefficiencies, and to improve maintainability” [Swanson 76]

Table 2-1: Different definitions of IT-system maintenance

Despite the variations in the definitions, there looks to be an agreement that IT-system maintenance has the intension to change an existing IT-system. The disagreements lie in which changes are to be included in IT-system maintenance, for example if a development of a new subsystem is IT-system-maintenance or not.

In this investigation we define this type of maintenance as traditional maintenance. Traditional maintenance is divided into 3 typed: corrective, adaptive and perfective maintenance [Swanson 76]. We will use a similar, but modified version introduced in [Krogstie 94].

Swanson's 3 types of maintenance can be described like this:

<u>Corrective maintenance</u>
Performed to identify and correct processing failures, performance failures and implementation failures.

<u>Adaptive maintenance</u>
Performed to adapt the software to it's changing technical surroundings.

<u>Perfective maintenance</u>
Performed to improve performance, alter or add new functionality, or improve further maintenance of the software.

Perfective maintenance can be divided into 2 types:

- Functional alteration includes changes to the software's functionality.
- Non-functional alteration includes changes done to improve quality factors like performance, resilience, and how easy it is to do maintenance on the system.

This grouping has been fundamental in many empirical studies. Preventive maintenance has also been used by some inquirers and describes the work that is done to a system to ease maintenance by restructuring the system. This type of maintenance is here included in the category non-functional perfective maintenance.

In addition to the traditional understanding of what is maintenance and what is development, we use the concepts functional maintenance and functional development:

<u>Functional maintenance</u>
Functional maintenance is described by [Krogstie 94] as: "Work made to keep up the functional coverage of the information systems portfolio of the organization". Functional maintenance includes adaptive-, corrective-, non-functional perfective maintenance, together with development of IT-systems which replace existing systems.

<u>Functional development</u>
Krogstie defines functional development this way: "Development or maintenance where changes in the application increase the functional coverage of the total information systems portfolio of the organization. This includes development of new systems which covers areas which are not covered earlier by other systems in the organization, and also includes functional perfective maintenance." [Krogstie 94].

In [Krogstie 94] it is indicated that the classification above gives a better indication of how well the IT-departments work supports the organization. Figure 2-1 shows the connection between functional maintenance, functional development, traditional maintenance and traditional development.

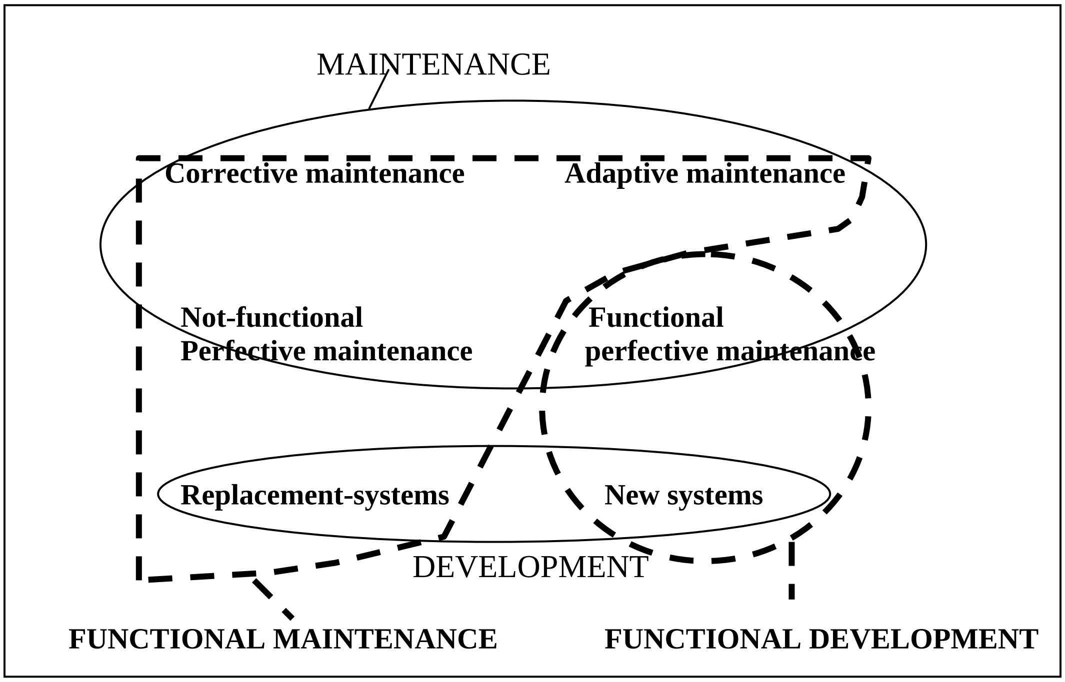

Figure 2-1: Connection between functional maintenance, functional development, traditional maintenance and traditional development [Krogstie 94]

As we can see from Figure 2-1, we divide development into two categories; replacement-systems and new systems. Replacement-systems are systems that replace older systems, and offer at least the same functionalities as the older system. New systems cover functional areas that have not been covered in the existing systems.

2.2 Previous investigations

There have been some investigations on maintenance-work performed in the IT-departments. They have among other things looked at factors that are related with maintenance. In Table 2-2 we gives an overview of some of the investigations, sorted increasing on share maintenance.

Maintenance (%)	Investigation	Year
49	[Lientz and Swanson 80]	1977
56	[Jørgensen 94]	1994
58	[Nosek and Palvia 90]	1990
59	[Krogstie 94]	1993
63	[Martiniussen 96]	1996
66	[Yip 95]	1995
66	[Dekleva 92]	1990
73	[Holgeid 99]	1998

Table 2-2: Previous investigations and share maintenance

The large spreading in share maintenance can be explained by: what sort of software has been examined, measure-errors and other source of error. One reason can also be that different definitions of maintenance have been used. For example in the investigations of [Yip 95] and [Dekleva 92] user-support was included as maintenance.

In Table 2-2 we can see different investigation running back to 1977. Four of these investigations have overlapping questions with our investigation, and are therefore more interesting. Below we will give a short presentation of these investigations, and we will later compare our results to theirs.

2.2.1 Lientz and Swanson

In 1977 Lientz and Swanson performed a study on maintenance in IT-systems. The 487 respondent organizations were picked out from the DPMA's (Data Processing Management Association) member registry. From a population of 7000 they picked out a random sample of 2000 leaders from different organizations. From the 2000 organizations they got a response-rate of 24,6%.

In Lientz and Swanson's investigation, traditional maintenance made out 50% of the total work done by the system-developers. This was similar to an investigation done 2 years earlier. One interesting observation was that organizations that organized maintenance-work and development separately used less time on maintenance than organizations that did not organize maintenance-work and development separately. Corrective maintenance amounted to 21,7% of the total time used on maintenance, adaptive maintenance 23,6% and perfective maintenance 51,3%. Functional perfective maintenance amounted to 41,8% of the total time used on maintenance. One of the conclusions of this investigation was that the size of the systems increased with their age. Usually also the maintenance-work increased with the age of the systems. Below we have listed the six biggest problem areas within maintenance of IT-systems found in this investigation:

1. The users knowledge
2. The programmer's efficiency
3. The quality of the product
4. The programmer's accessibility
5. Hardware-requirement

6. The system’s resilience

Lientz and Swanson found out that when the developers used test-data generators, structured programming and structured review of programming-code, the quality of the product was higher, and less time was needed to maintain these systems.

2.2.2 Nosek and Palvia

Nosek and Palvia’s investigation was a follow-up investigation to [Lientz and Swanson 80]. From 240 organizations that received the questionnaire, 52 (22%) answered. The respondent organizations had (median-values) 550 employees, IT-departments with 10 full-time employees whereof 6 within development and maintenance.

58% of the work-time was used on maintenance, while 35% was used on development. This is a relatively large increase compared to [Lienz and Swanson 80]. Functional perfective maintenance was reported to make up 42% of the total maintenance-work.

The biggest problem areas reported in this investigation are listed below:

1. The programmer’s accessibility
2. The programmer’s efficiency
3. Technical environment/ platform
4. The users knowledge
5. The quality of the product

2.2.3 Krogstie

In 1993 Krogstie performed an investigation on IT-systems development and maintenance in Norwegian organizations [Krogstie 94, Krogstie and Sølvberg 94, Krogstie 95]. This investigation was based on answers from 52 respondents, who were sampled out from the member-registry of The Norwegian Computer Society.

Krogstie reported that maintenance made up 59% of the total work within development and maintenance. Krogstie reported that following factors were connected with the increasing amount of maintenance-work:

- Smaller IT-departments
- Fewer developers
- Fewer developers with formal education within IT
- Higher average age of the application portfolio
- Larger problems with the users knowledge

Krogstie wrote that “The area which most of all differentiate this investigation from previous investigations is the distinction between functional development and maintenance, and it is our belief that the amount of functional maintenance is a better indicator on the efficiency in the information systems support in an organization than the amount of maintenance which are often used" [Krogstie 94].

Below we provide a list of factors that Krogstie reported to be connected with the increase in the amount of functional maintenance:

- Smaller IT-departments
- Fewer developers
- Fewer developers per main-system
- Smaller IT-departments per amount of end-users
- Less average work-experience among the developers compared to the application portfolios age
- Larger amount of new developments are replacement-systems
- Less structured maintenance-process (fewer organizing-controls used)
- Lack of a complete development-methodology in the organization

All of the factors above correlated with the size of the organizations, especially with the number of developers in the IT-departments, and number of end-users.

2.2.4 Holgeid

In 1998 Holgeid performed an investigation on IT-systems development and maintenance in Norwegian organizations. This was a similar investigation to the one performed by Krogstie in 1993.

This questionnaire was sent out to 470 organizations picked from the member registry of The Norwegian Computer Society. The results in this investigation are based on the answers from 53 organizations.

Holgeid reported 73% traditional maintenance when we look at maintenance and development alone. This was an increase compared to the investigation 5 years earlier. He reported that this increase could be partly explained by the Y2K problem.

Work related to the development of IT-systems functionality mad up 38%, while maintaining the existing functionalities made up the remaining 62%. This was a notable change from what was reported 5 years earlier.

Holgeid reported that the size of the organizations correlated with the amount of maintenance-work being done, especially number of employees. Organizations with many employees had significantly less maintenance-work compared to organizations with fewer employees.

The developer's competency was also reported to have influence on the amount of maintenance-work. Where organizations with developers with higher education, and organizations with developers with more experience had significantly less maintenance-work than the remaining organizations.

Holgeid reported also that the size of application portfolio influenced the amount of maintenance-work. Organizations with complex application portfolios were reported to have significantly less maintenance-work than organizations with less complex application portfolios.

2.3 Lifecycle-models and maintenance

In this investigation we do not look much at how different lifecycle-models affect maintenance, but it is still interesting to look at maintenance in connection with different lifecycle-models so we can place the maintenance-problems in a larger view.

Different lifecycle-models have different views on what system-development is and how this work should be performed. Below we will look at some of the lifecycle-models in connection with this investigation.

2.3.1 "Code-and-fix" model

The "code-and-fix" model is the first model for software-development. It consists of two steps: 1) write some code, and 2) fix the errors in the code. With this model you write the code without taking demands, design, testing and maintenance into consideration. [Boehm 88] described three primary problems that represent this model:

1. After some changes the code becomes unstructured [Lehman and Belady 85], which makes any future changes expensive.
2. The systems were often not used because the user-demands were not considered before the implementation.
3. Lack of preparation for testing resulted in that it was expensive to make changes in the code.

The first and third point involves that the costs of maintenance increased with time, and the second point can involve huge loads of work to meet the user-demands.

2.3.2 Waterfall model

This model came as an answer to the "code-and-fix" model. Here the lifecycle is divided into distinct phases.

The waterfall model eliminates many of the problems in the "code-and-fix" model, and it has become a basis for most of the programming-norms. According to [Boehm 88] one of the problems with the waterfall model is:

"A primary source of difficulty with the waterfall model has been its emphasis on fully elaborated documents as completion criteria for early requirements and design phases. For some classes of software, such as compilers or secure operating systems, this is the most effective way to proceed. However, it does not work well for many classes of software, particularly interactive end-user applications. Document-driven standards have pushed many projects to write elaborate specifications of poorly understood user interfaces and decision-support functions, followed by the design and development of large quantities of unusable code."

2.3.3 The evolutionary model

Problems described with the waterfall model resulted in the evolutionary model. This model aims to expand / develop already operational IT-systems over time. This model is suited in situations where the user can not explain what he wants before he sees it.

The problem with the evolutionary model resembles the one in "code-and-fix" model. It is easy to develop a lot of code which can be difficult to change because of the lack to take long-term architecture considerations. This could involve similar consequences for maintenance as described in chapter 2.3.1.

2.3.4 The transformation model

The transformation model was developed to try to fix the evolutionary model's problem with unstructured code. The intention with this model was to automatically convert a formal specification to a system that satisfies the specification. With the transformation model you do not change the code directly – you change the specification and then generate the code. With this solution we avoid the problem that the systems structure is getting more complex over time.

This model still has some of the problems we encountered in the evolutionary model; we assume that the system is flexible enough to changes that are not planned in advance. The users have also often problems understanding formal specification. It is not much help to have a formal specification if the user-demands are not accommodated.

2.3.5 The spiral model

The spiral model was developed based on the experience from the earlier models. This model is based on that system-development is made in iterations, and every iteration has a phase with risk evaluation.

The spiral model does not divide into development and maintenance. Boehm claims that this counteracts against that many people think of maintenance as low status-work [Boehm 88]. This diffuse division between development and maintenance is by others emphasized as negative since maintenance requires an understanding of the existing system, and often leads to re-testing and supervision of the system [Chapin 88, Harjani et al 92].

3 Research method

This report is based on the results of a survey investigation performed among a selection of Norwegian organizations. We will here take a closer look at what methods the questionnaire is based on. Chapter 3.1 will present the questionnaire, and chapter 3.2 will describe the analyses of the data.

3.1 Presentation of the questionnaire

The work with the questionnaire started in November 2002. The Norwegian Computer Society agreed to support the investigation with access to their member-records, while Simula Research Laboratory provided the financial and technical support.
The questionnaire was entered into an internet-based questionnaire-program called SESE, which is made by Simula and distributed in November 2003 to 200 organization members of The Norwegian Computer Society who were registered as the organizations contact-persons. The majority of these contact-persons were IT-leaders in the organizations. To secure that the organizations were of some size, a sample-criteria was used. The organizations were to have 3 or more employees. These criteria gave us a population of 800 organizations, and we picked out a random sample of 200 organizations. Both the population and selection process was similar to the one in [Holgeid 99] and [Krogstie 94], and it is therefore possible for us to compare our results to the ones in these investigations.

Before the questionnaire was distributed to the organizations, a pilot study was done. This helped us to correct some of the questions so that they were easier to understand.

The questionnaire was distributed along with a letter of information and a request from The Norwegian Computer Society to answer the questionnaire. To gain a highest possible response rate, the respondents were guaranteed confidential treatment of the answers, and an offer to get the final result sent to them. The respondents were also given 500kr, either to themselves or the organization.

The deadline to answer the questionnaire was first set to 8. of December. When the deadline had past, we sent out reminders to the organization that yet had not answered. After the deadline for the reminder had past, we decided to send out questionnaires to a new selection of 50 organizations, based on a random sample from the 600 not elected in the first round. This meant that we had now sent out questionnaires to 250 organizations. This was half as many as in [Holgeid 99], but because of the higher reward for answering, we hoped to get a higher answer-percent.

The number of fully answered questionnaires was 46. In addition to this we got 8 not fully answered questionnaires we could use, since they included data for the major areas of the investigation.

The total number of answered questionnaires we could use was 54. Totally there were 247 organizations that received the questionnaires (3 were returned due to wrong address). This gives us a response rate of 21,6%.

Compared to [Holgeid 99] the number of questionnaires was almost the same (53). We have a higher response-percent than in [Holgeid 99] (17,4%) because of the lower number of dispatched questionnaires in our investigation.

We found that the 54 usable questionnaires we had received made a relatively good basis for further analyses.

3.2 Analysis

The results were taken from the internet-based questionnaire-program and coded into SPSS. To minimize the possibility of coding-error, we verified several times that the coded data was the same as the collect data from the questionnaires. Verification was done by other people than the one doing the original coding. SPSS was used to perform all the analysis of the data, and we have used the same statistical-methods (hypothesis-testing and correlation analysis) as in [Lientz and Swanson 80], [Nosek and Palvia 90], [Krogstie 94] and [Holgeid 99]. In connection with hypothesis-testing we used these two lines of action:

- Testing for equality of the average between two separate variables.
- Split the sample according to the sum of a variable, and test for likeness of the average of another variable in each of the sub-samples. When the results of the analysis are significant, the correlation analysis is also presented.

The hypothesis that are tested are partially based on former questionnaires like [Lientz and Swanson 80], [Nosek and Palvia 90], [Krogstie 94] and [Holgeid 99]. The hypothesis in connection with functional development and functional maintenance are mainly based on [Krogstie 94] and [Holgeid 99].

In addition to the hypothesis we have compared central variables from [Holgeid 99] with similar variables in our questionnaire.

3.3 Descriptive statistics

The results from descriptive statistics can tell us how the data in a sample is distributed, and which tendencies in the data it is possible to trace.

In the report we have used the following traditional descriptive statistical measures:

- Average
- Median
- Mode

- Minimum- and maximum-values
- Standard Deviation

In addition to these traditional statistical measures, we use more advanced methods to test how the data is distributed. To investigate if a variable is normally distributed, we use two different measures: skewness and kurtosis. Skewness shows us the distributions degree of symmetry. A symmetric distribution has skewness equal 0. While a positive value tells us that the distribution is pulled to the left, and a negative value tells us that the distribution is pulled to the right. Kurtosis measures if a distributions center (or top) is higher or smaller than the one for a normal distribution, and if the “tales” to the left and right are shorter or longer compared to a normal distribution. A normal distribution has kurtosis equal to 0. A distributions skewness and kurtosis should have an absolute-value less than 2, if the distribution is to be called normal [Norusis 92].

In addition to skewness and kurtosis, we have also used “Lillefors-test” and “Shapiro-Wilks-test” to test for normality [Norusis 92].

3.3.1 Measure-level

In the questionnaire there are carried out several measures related to maintenance of the systems. The measure-scale which is used determines measure-level, and the measure-level determines which statistic method should be used on the data. The higher the measure-level, the stronger statistic method can be used. A stronger statistic method is a method which will more often give us a statistic significant result than a weak method, with the same data. The measure-level is traditionally divided into 4 levels: nominal, ordinal, interval and ratio [Ask 94].

Variables on the nominal-level will be tested with the weakest statistic methods, and the possibility of finding statistic significance is reduced. These characteristics distinguish the nominal-level [Ask 94]:

- No natural sequence between the different measuring values.
- The unit on the measure-scale is not equal for all measuring values.
- The distribution will never be normally distributed.
- The scale has no natural zero point.

Variables that have either the value, YES or NO, are typically on nominal-level.

Variables on the ordinal-level have these characteristics:

- The measuring values have ranking between themselves.
- The unit on the measure-scale is not equal for all measuring values.
- The distribution can be normally distributed
- The scale has no natural zero point.

A typical variable on ordinal-level will for instance have the measuring values Low, Average and High.
The interval-level have these characteristics:

- The measuring values show the sequence between elements.
- The unit on the measure-scale is equal for all measuring values.
- The distribution can be normally distributed. The variables are continuous, and therefore satisfy the foundational demand for a normal distribution.
- The scale has no natural zero point.

The ratio-level, is the highest level a variable can have. The difference between the ratio-level and the interval-level is that the measuring-scale has a zero point for variables in ratio-level.

3.4 Hypothesis testing

The purpose of the hypothesis testing is to come to conclusions regarding the parameters of the population based on observed results in a random sample. The hypothesis testing in this report will mainly consist of comparing two or more sub-samples to see if the difference between the sub-samples is big enough to reject the null hypothesis. Rejecting a null hypothesis is not the same as saying that the alternative hypothesis is right, but it will give the alternative hypothesis more support. The reason for that rejecting the null hypothesis is not a final proof that the alternative hypothesis is right, is because there may be other variables that influence the relation.

When testing null hypothesis we use either t-tests, or non-parametric tests. T-tests are stronger than non-parametric tests. The t-tests assume that the data in the sample are normally distributed. If this is not the case, we can use non-parametric test like Mann-Whitney test (also called Wilcoxon test). The purpose of non-parametric tests is that the measure-values that differ a lot from the average values will not get a disproportionately big influence on the results.

For a variable to be normally distributed, the measure-scale has to be continuous. This is why only variables from the interval- and ratio-level can be normally distributed. Variables on the nominal- and ordinal-level are discrete, and will never be normally distributed, even though variables on the ordinal-level can be approaching a normal distribution when the measure-level is fine enough.

T-tests can however be used on samples that are not normally distributed, because the average of the samples will be normally distributed and group themselves around the average for the population. This phenomenon is explained in the rule for normal distribution [Wonnacott et al 90].

If the observed significance level is small enough, the null hypothesis is rejected. When validating a null hypothesis (H_0) there are two mistakes we can make:

1. Reject H_0 when it is correct.
2. Accept H_0 when it is not correct.

The significance level (p) gives us the probability to make mistake number 1. The probability to reject H_0 when it is not correct should be large, so mistake number 2 does not occur.

3.5 Correlation coefficient

Pearson's correlation coefficient (r) is often used. The absolute value of r indicates the strength to the linear relation between two variables with 1 as the highest value – which occurs when all points falls on exactly the same line. A correlation coefficient equal to 0 indicates that there is no linear relation between the variables. The variables can still have a strong relation, even though a small correlation coefficient, but it is then not a linear relation.
If a correlation analysis between two variables x and y gives us a positive correlation coefficient and a significant p-value, then the linear relation between the two variables is that y increase when x increase. If the correlation coefficient is negative, the opposite will happened – y decrease when x increase.

Pearson's correlation coefficient can only be used if the variables are at least on the interval-level, and are normally distributed. This will say that variables on the nominal- and ordinal-level are not applicable for such analysis [Wilson 71]. There are some researchers who suggest making correlation analysis with variables on ordinal-level [Boyle 70, Nie and more 75, Labovitz 70]. [Guilford 78] claims that the normality-demand can be reduced, but that the sample is to be unimodal and symmetric. [Bergersen 90] says that the number of measure-sets should be over 30, and to consider the relation between two variables as important, the r-value should at least be 0,3 with an significance level of 0,01. In this context the significance level can be explained as the probability that there is no linear correlation-relation between the variables. [Lientz and Swanson 80] used a r-value larger or equal 0,1 with significance level of 0,01 and 0,001, while [Swede and Vliet 94] suggests to use a r-value larger than 0,25 with a significance level of 0,05. We will do like [Krogstie 94] and [Holgeid 99] and use the suggestion of [Swede and Vliet 94].

For variables on ordinal-level or interval- and ratio-level that are not normally distributed, we use Spearman's correlation coefficient (S). Spearman's correlation coefficient is a non-parametric version of Pearson's correlation coefficient.

4 Hypothesis

We will in this chapter present the hypotheses. These hypotheses will be tested in chapter 6. The hypotheses are mainly built around relative sizes as: many, fewer, small, large. What these relative sizes represent appears in chapter 6. We use these relative sizes in the hypotheses since we compare the sub-samples which are decided from the samples median.

The hypotheses are organized in different groups: maintenance/development, size and type of organization, importance of IT, internal competency, complexity of the portfolio, organizing, use of methods and use of tools.

4.1 Maintenance/Development

H1: There is no difference in the amount of work-load used on maintenance and development, when we only look at maintenance and development.

H2: There is no difference in the amount of work-load used on maintenance and development.

H3: There is no difference between the work-load used on functional maintenance and traditional maintenance, when we look at development and maintenance only.

H4: There is no difference between the work-load used on functional development and traditional development, when we look at development and maintenance only.

H5: There is no difference between the work-load used on functional development and functional maintenance, when we look at development and maintenance only.

4.2 Size and type of organization

H6: There is no difference in the amount of maintenance-work between organizations with many employees and organizations with fewer employees.

4.3 Importance of IT

H7: There is no difference in the amount of maintenance-work between organizations where IT is of big strategic importance and organizations where IT is of less strategic importance.

H8: There is no difference in the amount of maintenance-work between organizations in which the size of the IT-department compared to the total number of employees is large and the organizations where the size of the IT-department compared to the total number of employees is small.

H9: There is no difference in the amount of maintenance-work between organizations in which there are many system-developers in proportion to total number of users, and organizations with few system-developers in proportion to total number of users.

H10: There is no difference in the amount of maintenance-work between organizations with large IT-budgets in proportion to total number of employees, and organizations with small IT-budgets in proportion to total number of employees.

4.4 Internal competency

H11: There is no difference in the amount of maintenance-work between organizations with system-developers who have in average many years of experience in the IT-department, and organizations with system-developers who have shorter experience.

H12: There is no difference in the amount of maintenance-work between organizations with system-developers who have in average many years of experience in the IT-department in proportion to the main-systems average age, and organizations with system-developers who have short experience in proportion to the average age of the main-systems.

H13: There is no difference in the amount of maintenance-work between organizations with many hired IT-consultants within system-developing per system-developer, and organizations with few hired IT-consultants within system-developing per system-developer.

4.5 Complexity of the portfolio

H14: There is no difference in the amount of maintenance-work between organizations with many main-systems and organizations with fewer main-systems.

H15: There is no difference in the amount of maintenance-work between organizations with many users and organizations with fewer users.

H16: There is no difference in the amount of maintenance-work between organizations with main-systems with high age average, and organizations with main-systems with low age average.

H17: There is no difference in the amount of maintenance-work between organizations with main-systems which are highly dependent on data from other systems, and organizations with main-systems which are less dependent on data from other systems.

H18: There is no difference in the amount of maintenance-work between organizations with many different system-configurations and organizations with fewer different system-configurations.

H19: There is no difference in the amount of maintenance-work between organizations that use many different programming-languages, and organizations that use fewer different programming-languages.

4.6 Organizing

H20: There is no difference in the amount of maintenance-work between organizations where the maintenance-workers are organized differently from the developers and organizations where there is no such difference.

H21: There is no difference in the amount of maintenance-work between organizations where maintenance is often performed by the people who developed the system, and organizations where maintenance is rarely performed by the people who developed the system.

4.7 Use of methods

H22: There is no difference in the amount of maintenance-work between organizations that use methodologies in the IT-systems lifecycle, and the organizations that do not use this.

4.8 Use of tools

H23: There is no difference in the amount of maintenance-work between organizations that use system development tools, and the organizations that do not use this.

H24: There is no difference in the amount of maintenance-work between organizations that use system maintenance tools, and the organizations that do not use this.

5 Descriptive results

In this chapter we will present the descriptive data based on the questionnaire.

The terms used in the tables are from the questionnaire, and will therefore be in Norwegian, like in the questionnaire.

5.1 Respondents

Under this chapter we will look at the respondents' background and get an overview of their experience with IT.

The questionnaire was send out to IT-leaders, and we expected them to answer it themselves, since they probably have the best overview of the relevant data for the inquiry form.

Category	Frequency	Percent
Leder	44	81,5
Prosjektleder	3	5,6
Systemutvikler	7	13,0
Total	**54**	**100,0**

Table 5-1: Respondents title

Table 5-1 indicates what job title the respondents have. We can se that 87,1% of the respondents are in the category "Leder" (81,5%) or "Prosjektleder" (5,6). It is of interest to verify that it is mostly IT-leaders among the respondents, since the questionnaires we compare us to are also mainly based on the response of IT-Leaders (94% leaders in [Krogstie 94] and 90.6% in [Holgeid 99]).
Other questionnaires have found out that leaders might interpret the maintenance-problem differently from the employees who carry out these tasks ([Jørgesen and Maus 94]). This can lead to different results than if the respondents were mainly developers/maintenance-workers. We will not look at this factor in our investigation, since the questionnaires we compare us to are also mainly based on the response of IT-leaders.

Category	Frequency	Percent
Fast	53	98,1
Konsulent	1	1,9
Midlertidig	0	0

Table 5-2: Respondents appointment-status

People with a temporary employment in a company are likely to know less about the situation in the company then people with a permanent employment. This is quite important, and makes it interesting to read the results in Table 5-2 which say that 98,1% of the respondents are permanently employed. We can compare this to the results in [Krogstie 94] and [Holgeid 99] where there was also a result of only 1,9% temporary employed, and the rest of the respondents was permanently employed.

N	Min	Max	Sum	Mean	Median	SD
54	3,0	35	782	14,5	15	7,5

Table 5-3: Years IT-experience

How many years of IT-experience the respondent have is also of interest, because people with little experience are more likely to answer based on what they have learned is problematic with maintenance – compared to respondents with more experience who will base their answers more on what problems they have experienced themselves.
In Table 5-3 we can see the respondents IT-experience, which says that the average IT-experience among the respondents is 14,5 years. This is about the same as in [Holgeid 99] (14,2 years) and a little lower than [Krogstie 94] (16,7years).

5.2 Organizational profile

Under this chapter we will take a closer look at organizations the respondents represent.

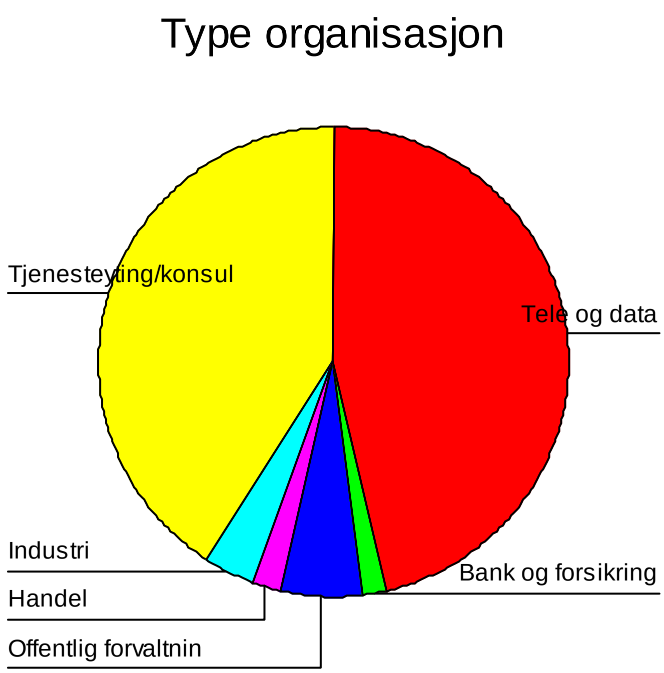

Figure 5-1: Type of organization

Type of organization	Frequency	Percent
Tele og data	25	46,3
Bank og forsikring	1	1,9
Offentlig forvaltning	3	5,6
Handel	1	1,9
Industri	2	3,7
Tjenesteyting/konsulentvirksomhet	22	40,7
Total	**54**	**100,0**

Table 5-4: Type of organization

From Figure 5-1 and Table 5-4 we can see that the majority of the respondents work within "Tele og data" (46,3%) and "Tjenesteyting/konsulentvirksomhet" (40,7%). The organization types were in earlier questionnaire regrouped to production-organizations and service-organizations. If we do the same regroup we get the result that 96,3% of the respondent organizations are service-organizations, while 3,7% are production-organizations. This continues the growth of service-organizations we see from comparing [Holgeid 99] (79,2%) and [Krogstie 94] (61,5).

Category	Frequency	Percent
Absolutt	43	79,6
Stor	7	13,0
Tildels	2	3,7
Lite	1	1,9
Nei	1	1,9
Total	**54**	**100,0**

Table 5-5: It of strategic importance

Table 5-5 shows in which degree the respondents think IT is of strategic importance for their organization. The majority (92,6%) consider IT of "Absolutt" strategic importance (79,6%) or of "Stor" strategic importance (13%). This picture does not change much when we only look at respondent who are leaders.

N	Min	Max	Sum	Mean	Median	Mode	SD
54	3	2400	9789	181	27	70	480,92

Table 5-6: Number of employees

Table 5-6 shows us the total employees in the organizations, and gives us an indication on their size. The average number of employees among the respondent organizations is 181 and the median of 27. This is less than in both [Holgeid 99] (mean = 656 and median = 160) and [Krogstie 94] (mean = 2347 and median = 555). Since we in earlier investigations have found out differences related to the size of the responding organizations, we should keep this in mind when comparing our results with these.

Mill NOK	Budsjett 2003		Budsjett 1998		Budsjett 1993	
	Frequency	Percent	Frequency	Percent	Frequency	Percent
Mer enn 50 mill	4	7,4	9	17,0	4	9,3
Mellom 40 og 50 mill	2	3,7	2	3,8	1	2,3
Mellom 30 og 40 mill	0	0,0	2	3,8	1	2,3
Mellom 20 og 30 mill	1	1,9	2	3,8	2	4,7
Mellom 10 og 20 mill	4	7,4	8	15,1	5	11,6
Mellom 1 og 10 mill	28	51,9	18	34,0	13	30,2
under 1 mill	15	27,8	12	22,6	17	39,5
Total	**54**	**100,0**	**53**	**100,0**	**43**	**100,0**

Table 5-7: IT-departments yearly budget given in millions NOK

In Table 5-7 we have another way to see the organizations size and how important IT is for the organization. Table 5-7 shows us the IT-departments yearly budget for 2003, 1998 and 1993 given in millions NOK. We can se that in 1993, 1998 and 2003 the majority of the responding IT-departments have budgets up to 10mill NOK. From the budget for 2003 we can also see that there are now more organizations with an IT-department budget between 1 and 10mill NOK than in earlier years. From 1998 to 2003 the percentage of organizations with IT-budgets over 50mill NOK has decreased. This is obviously related to the size distribution of the respondents as discussed above.

5.3 Distribution of labor

This chapter will present the distribution of labor between development- and maintenance work.

Activity	N	Min	Max	Mean	Median	SD
Feilretting	52	0	30	8,7	10,0	6,1
Adaptiv	52	0	22	7,2	5,0	4,9
Funksjonell	52	0	50	12,5	10,0	11,3
Ikke-funksjonell	52	0	30	7,5	5,0	6,0
Erstatning	52	0	40	9,7	10,0	8,4
Nyutvikling	52	0	62	12,2	10,0	13,8
Drift	52	2	70	23,1	20,0	16,4
Brukerstøtte	52	0	50	16,8	12,0	13,2
Annet	53	0	50	2,3	,0	7,4
Total andel utvikling	**52**	**0**	**81**	**21,9**	**20,0**	**17,8**
Total andel vedlikehold	**52**	**5**	**80**	**35,9**	**31,0**	**15,6**

Table 5-8: Distribution of labor in percent of IT-departments total time consumption

Table 5-8 shows us the distribution of labor in percent of IT-departments total time consumption. The 4 first categories in the table form traditional-maintenance, while development of new systems classifies under traditional-development. From Table 5-8, we can see that IT-departments on average use 35,9% of the time on maintenance, while 21,9% is used on development. In [Holgeid 99] we can see that 41,4% of the time was used on maintenance and 17,1% on development. We can see there are a few percent that have moved from maintenance to development. In our questionnaire “Drift” is at 23,1%, almost exactly the same as in [Holgeid 99] (23,0%), while “Brukerstøtte” (16,8%) has

decreased a little compared to [Holgeid 99] (18,6%). Compared to [Lientz and Swanson 80] development came to 43% and maintenance 49%, while other work only came to 8%. [Nosek and Palvia 90] reported 35% development and 58% maintenance.
Out of this we can see that in all of these questionnaires maintenance have taken up more labor time than development.

Development of new functionalities in existing systems comes to 12,5% and error recovery productive systems 8,7%. This is less than in [Holgeid 99] (15,2% and 12,7%).

Grade of accuracy	**Frequency**	**Percent**	**Vedlikehold %**
Rimelig nøyaktig	2	3,7	23,5
Et grovt estimat	25	46,3	39,0
En best mulig gjetning	23	42,6	34,1
Total	**50**	**92,6**	

Table 5-9: Quality of Table 5-8

To get an impression of how accurate the percentage in Table 5-8 is, we specifically asked on which foundation the answers was based on. The results we find in Table 5-9 shows that as many as 42,6% of the answers are based on “En best mulig gjetning”, while 46,3% of the respondents answered “Et grovt estimat”. Only 3,7% of the answers were based on good data.

When we look at development and maintenance isolated, development comes to 34,1% while maintenance comes to 65,9%. Compared to [Holgeid 99] (development = 27,1% and maintenance = 72,9%) and [Krogstie 94] (development = 41,4% and maintenance = 58,6%).

Functional development in our questionnaire comes to 38,9%, and the functional maintenance 61,1%, when we look at them isolated. This is very close to the results from [Holgeid 99] (functional development = 37,7% and functional maintenance = 62,3%), while [Krogstie 94] reported 56% functional development and 44% functional maintenance.

5.4 IT-department

This chapter will describe the IT-departments of the respondent organizations to give us an impression of them. To do this we will look at IT-department size and the employees experience and education.

N	**Min**	**Max**	**Sum**	**Mean**	**Median**	**Mode**	**SD**
54	,2	70	526,6	9,8	5	1	14,1

Table 5-10: IT-departments employees

The size of the IT-department can be described by the amount of employees in the department. Table 5-10 shows us the employees in the IT-department, recalculated to amount of full-time employees. In our questionnaire the mean in an IT-department is 9,8 employees, this is a little lower than in [Holgeid 99] (10,9). [Lientz and Swanson 80] (mean = 45,4 employees), [Nosek and Palvia 90] (mean = 178 employees), [Swanson and

Beath 89] (mean = 95 employees) and [Krogstie 94] (mean = 24,3 employees). Both our questionnaire, [Holgeid 99] and [Krogstie 94] reported of a much smaller number of IT-department employees than the mentioned American questionnaires. If we just compare [Krogstie 94] (24,3), [Holgeid 99] (10,9) and our questionnaire (9,8) it appears that the number of employees in IT-departments is decreasing. When we compare number of employees in IT-departments in our questionnaire with [Holgeid 99], it is important to remember that in our questionnaire we only had an average of 181 employees in an organization, while the number was 656 in [Holgeid 99]. The IT-department amount to, in average, 5,4% of the organizations total number of employees, while In [Holgeid 99] this number was 1,7%. This is an increase from 1,0% in [Krogstie 94] to 1,7% in [Holgeid 99] and 5,4% in our.

N	Min	Max	Sum	Mean	Median	Mode	SD
54	0	45	221,7	4,1	1,5	0	7,0

Table 5-11: Number of system-developers in the IT-department

From Table 5-11 we can se that in average there are 4,1 full-time system-developers employed in each IT-department. In average the system-developers amount to less then half of the total number of employees in the IT-department (42%). The amount of system-developers compared to the amount of employees in the IT-department is reported to be relatively equal to other questionnaires. In [Lientz and Swanson 80] there was in average 38% system-developers in the IT-department, in [Nosek and Palvia 90] 43%, in [Krogstie 94] 39%, and in [Holgeid 99] 42%. The proportionality between the number of system-developers and the total number of employees in an IT-department can not be used to explain the differences in other labor than development and maintenance as mentioned in chapter 5.3.

Years	N	Min	Max	Sum	%	Mean	SD
0-1 år	51	0	2	12	5,8	,2	,6
1-3 år	51	0	4	44	21,2	,9	1,3
3-6 år	51	0	45	95	45,7	1,9	6,3
6-10 år	51	0	8	37	17,8	,7	1,6
Mer enn 10 år	51	0	5	20	9,6	,4	1,1

Table 5-12: System-developers divided in groups of how long they have been working in the organizations IT-department

Table 5-12 shows us the system-developers divided in groups of how long they have been working in the organizations IT-department. We can see that 45,7% (95) of the system-developers have been working in the same IT-department between 3 and 6 years. Only 9,6% have been working longer than 10 years in the same IT-department, this is a big decrease from the result in [Holgeid 99] (29,3). From the results in [Holgeid 99] we expected the age-group 3-6 years to increase drastically, which it also did. [Holgeid 99] 3-6 years = 14,4%, our 3-6 years = 45,7%. This is because of the high percentage in the lower groups (0-1 years and 1-3 years) in [Holgeid 99].

Years	N	Min	Max	Sum	%	Mean	SD
0-1 år	51	0	2	3	1,4	,1	,3
1-3 år	51	0	3	13	6,2	,3	,7
3-6 år	51	0	45	86	41,0	1,7	6,3
6-10 år	51	0	10	46	21,9	,9	1,9
Mer enn 10 år	51	0	10	62	29,5	1,2	2,2

Table 5-13: System-developers divided in groups of total experience

The system-developers total experience is presented in Table 5-13. We can see that only 29,5% (62) of the system-developers have more than 10 years of total experience compared to 49,7% (119,5) in [Holgeid 99]. We can also see that the number of system-developers with more than 6 years of total experience is lower in our questionnaire than in [Holgeid 99]. Our 51,4% (108), [Holgeid 99] 66,3% (159,5).

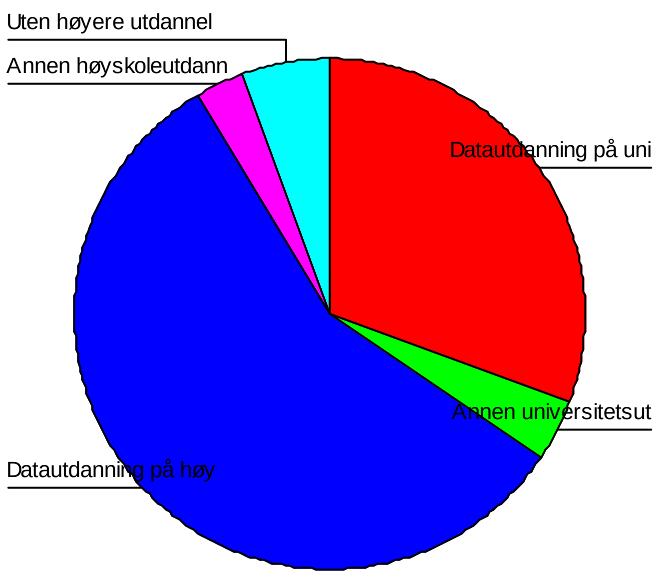

Figure 5-2: The system-developers average education-background

Utdannelse	Vår %	[Holgeid 99] %	[Krogstie 94] %
Datautdannelse på universistetsnivå	30	21	38
Annen universitetsutdannelse	4	8	17
Datautdannelse på høyskolenivå	57	31	20
Annen høyskoleutdannelse	3	6	12
Ingen formel utdannelse	6	34	13

Table 5-14: Percentage distribution of the system-developers education-background

There is written in [Vessey and Weber 83] that the system-developers qualifications are said to be related to efficient carrying out of maintenance and development. We used education-background as a measure of the qualifications. In Figure 5-2 and Table 5-14 we can see that as much as 87% of the system-developers have a higher education within computers. Compared to [Holgeid 99] (52%) and [Krogstie 94] (58%) this is very high. At the same time we can se that the number of system-developers without any formal education has decreased drastically. From 34% reported in [Holgeid 99] to 6% in ours.

Statistics	N	Min	Max	Sum	Mean	Median	Mode	SD
Datautdanning på universitetsnivå	53	0	11	70,5	1,3	0	0	2,5
Annen universitets utdanning	53	0	3	9,0	,2	0	0	,6
Datautdanning på høyskolenivå	53	0	40	130,5	2,5	1	0	5,7
Annen høyskoleutdannelse	53	0	3	7,0	,1	0	0	,5
Uten høyere utdannelse	53	0	3	13,0	,2	0	0	,6

Table 5-15: The system-developers education-background

In Table 5-15 we see a more detailed outline of the system-developers education-background. We can see that in average there are most system-developers with computer education at college level (mean = 2,5). The highest number, in one organization, of system-developers with computer education at college level was 40 out of totally 45 system-developers.

N	Min	Max	Sum	Mean	Median	Mode	SD
50	,00	6	36,7	,7	0	0	1,4

Table 5-16: Hired IT-consultancy services

Many organizations hire consultants for developing new system or when the systems need maintenance. In this questionnaire we have not explored what the hired IT-consultants do, but from Table 5-16 we can see that there is in average 0,7 (recalculated to full-time employees) consultants in an organization. This is lower than in [Holgeid 99] (2,7). This decrease can be a result of the size of the respondent organizations. Smaller organizations do not use consultants as much as the bigger ones. 56% of the organizations do not use consultancy services while the 44% remaining do. This shows that more than half of the organizations do not use consultancy services, this is higher than in [Holgeid 99], where 30,2% did not use consultancy services while the remaining 69,8% did.

Year	Formally		Informally		No		N
	N	%	N	%	N	%	
2003	9	17,6	5	9,8	37	72,5	51
1998	9	17,6	3	5,9	39	76,5	51
1993	4	9,8	2	4,9	35	85,4	41

Table 5-17: How system-developers working with maintenance are organized

There are different views whether it is good or not that system-developers working with maintenance are organized separate from the rest of the system-developers. [Reynolds 77] report that organizing the system-developers that work with maintenance separately creates communication-problems, while [Bronstein and Okamoto 81] says that this resolves the problem with conflicting priorities.

Table 5-17 shows us the extent of separate organization within system-developers performing maintenance. We can se that the percentage of organizations, that formally organize system-developers that perform maintenance differently than the rest of system-developers, have stayed at 17,6% (as in [Holgeid 99]). The number of organizations that organize informally (9,8%) has increased a little when we compare to [Holgeid 99] (5,9%). The percentage of organizations that do not organize system-developers, who perform maintenance differently, from the rest of the system-developers has decreased a little (72,5%), compared to [Holgeid 99] (76,5%), and follows also the decreasing tendency from [Krogstie 94] (85,4%).

Category	**Frequency**	**Percent**
Aldri	4	7,8
Sjelden	5	9,8
Av og til	17	33,3
Ofte	19	37,3
Alltid	6	11,8
Total	**51**	**100,0**

Table 5-18: Maintenance performed by same people that developed the system

In Table 5-18 we can see that in 7,8% of the organizations, maintenance is never performed by the same people who developed the system. On the other side we have 11,8% of the organizations where maintenance always is performed by the same people that developed it. It is also interesting to see that maintenance is “Ofte” performed by the same people that developed it in 37,3% of the organizations, and “Av og til” in 33,3% of the organizations.

Category	**Frequency**	**Percent**
Aldri	14	28,6
Sjelden	20	40,8
Av og til	10	20,4
Ofte	4	8,2
Alltid	1	2,0
Total	**49**	**100,0**

Table 5-19: Maintenance as training-activity

Maintenance is in 28,6% of the organizations never performed as a training-activity (Table 5-19). Only in 10,2% of the organizations maintenance is “Alltid” or “Ofte” performed as a training-activity. The majority of the organizations “Aldri” (28,6%), “Sjelden” (40,8%) or “Av og til” (20,4%) perform maintenance as a training-activity. In our questionnaire we have not asked what kind of training-activity, if any, the organizations use.

5.5 System portfolio

In this chapter we will look at the organizations system portfolio. We will also compare our result to [Holgeid 99], [Krogstie 94] and [Swanson and Beath 89].

N	Min	Max	Sum	Mean	Median	Mode	SD
53	0	15	237	4,5	3	3	3,1

Table 5-20: Running main systems

From Table 5-20 we can see that in average there are 4,5 main system running in an organization. There are in total 237 main systems running in the questionnaire. The number of running main systems in an respondent organization spreads from 1 at the lowest till 15 at the most. This is lower than in [Holgeid 99] (mean = 9,6, min = 1, max = 100).

We also asked the respondents about what kind of systems they consider to be main systems. Even though different systems are differently significant for the organizations, we could see that there was a common understanding to what a main system is. Here is a representative list of types of main systems the respondents use:

- economics/ wages
- sale/internet shops
- production systems
- control system for storage rooms
- administration systems
- information systems
- data storage

The main systems reported in this questionnaire are very similar to the ones reported in both [Holgeid 99] and [Krogstie 94]. This would mean that understanding of what a main system is has not changed much the past years.

N	Min	Max	Sum	Mean	Median	Mode	SD
53	0	1800	6076	115	16	10	342,3

Table 5-21: Internal users of the main systems (red in the figure 5-3)

From table 5-21 we can see that there are totally 6076 internal users of the main systems. We see a spreading from 0 till 1800 users. The average number of internal users for an organization is 115.

N	Min	Max	Sum	Mean	Median	Mode	SD
52	0	2000	10283	198	0	0	467,8

Table 5-22: External users of the main systems (green in the figure 5-3)

There are totally 10283 external users of the main systems (Table 5-22). The spreading in this category is from 0 till 2000 users, with an average of 198 external users of the main systems for each organization.

N	Min	Max	Sum	Mean	Median	Mode	SD
52	0	2005	16337	314	55	50	551,5

Table 5-23: Total overview of the users of the main systems

There are totally 16337 users of the main systems in this questionnaire (Table 5-23). Most frequently there are 50 users per organization, but there is a spreading from 0 to 2005 users. In average each organization has 314 users of their main systems. This is a decrease compared to [Holgeid 99] (498) and [Krogstie 94] (541), which is as expected since the size of the surveyed organizations also are less.

Years	N	Min	Max	Sum	%	Mean	Median	SD
0-1 år	51	0	8	47	20,4	0,9	1	1,3
1-3 år	51	0	10	86	37,4	1,7	1	1,7
3-6 år	51	0	5	61	26,5	1,2	1	1,2
6-10 år	51	0	4	17	7,4	0,3	0	0,8
Mer enn 10	51	0	8	19	8,3	0,4	0	1,3

Table 5-24: The main systems age distribution

The main systems age distribution is shown in Table 5-24. 20,4% (47) of the main systems are less than 1 year old, and 8,3% (19) of the main systems are older than 10 years. In [Holgeid 99] the numbers were 7,3% (37) less than 1 year old, and 18% (91) older than 10 years. In our questionnaire the most main systems 37,4% (86) are between 1-3 years old, while in [Holgeid 99] the most main systems were between 3-6 years old.

Category	N	Min	Max	Sum	%	Mean	Median	SD
Utviklet av dataavdelingen	52	0	7	47	22,6	0,9	0	1,3
Utviklet i brukeravdelingen	52	0	2	4	1,9	0,1	0	0,3
Utviklet av utenforstående	52	0	7	73	35,1	1,4	0	2,0
Pakkeløsning, store tilpasninger	52	0	5	25	12,0	0,5	0	1,1
Pakkeløsning, små tilpasninger	52	0	9	57	27,4	1,1	0	1,7
Komponentbasert	52	0	1	2	1,0	0,0	0	0,2

Table 5-25: The main systems development-process

From Table 5-25 we can see that most main systems are made by outsiders (35,1%), and only 1,9% by the user-group. This is a big change from [Holgeid 99] where most main systems were made by the data-department (26,8%) and user-groups (26,6%), while outsiders only made 21,9% of the main systems. Compared to [Krogstie 94], the changes are even bigger, where there was reported that 58% of the main systems were made in the organizations IT-department, while only 12% was made by outsiders. If we compare our reports to the ones in [Holgeid 99] and [Krogstie 94], we can see that there is now natural to buy main systems made by outsiders than it was earlier. One reason for this can be that organizations do not have the capacity to do it their selves, or that they find it easier to buy ready systems. It is also interesting to see that many organizations use systems made by outsiders, while there is few consultants attached to the organizations.

N	Min	Max	Sum	Mean	Median	Mode	SD
49	0	15	113	2,3	2	0	2,8

Table 5-26: Main systems dependence on data from other systems

It is likely that main system which are dependant on other system for their data are more time-consuming to maintain, especially when it comes to testing and reinstalling. In our questionnaire there is in average of 2,3 main systems dependent on data from other systems, this is lower than in [Holgeid 99] where the average was 5,9. We can say that just over 50%[1] of the main systems depend on data from other systems. This is a decrease of about 10% when we compare to [Holgeid 99] (60%), and about 24% decrease when compared to [Krogstie 94].

N	Min	Max	Sum	Mean	Median	Mode	SD
46	1	4	88	1,9	2	1	1,0

Table 5-27: Number of different configurations

We wanted get an impression of how maintenance can be affected by the multiplicity of computer-languages. Because of this we asked the respondents what kind of hardware and systems software they used, and number of systems which support these configurations.

In Table 5-27 we can see that there is an average of 1,9 different configurations in an organization, this is 1,1 configuration less than reported in [Holgeid 99]. The questionnaire includes 88 configurations, which spread from 1 to 4 configurations per organization.

N	Min	Max	Sum	Mean	Median	Mode	SD
36	0	5	72	2,0	2	1	1,3

Table 5-28: Number of different programming-languages

The respondents where also asked which programming-languages they used. In table 5-28 we can se the results, which tell us that in average there are 2,0 different programming-languages used in each organization. [Holgeid 99] reported in average 2,5 different programming-languages, while [Krogstie 94] reported 2,7.

Language	N	Number org.	%	N	Number of systems	%
Cobol	36	1	1,4	36	1	0,5
Assembler	36	1	1,4	36	1	0,5
C	36	9	12,9	36	26	12,5
C++	36	17	24,3	36	48	23,1
Java	36	19	27,1	36	62	29,8
4GL	36	12	17,1	36	28	13,5
Andre	36	11	15,7	36	42	20,2

Table 5-29: Use of programming-languages

From Table 5-29 we see that Java and C++ are the most used programming-languages, with respectively 27,1% (19) and 24,3% (17) using them. They are also the programming-languages which most programs are made in (Java = 68 (29,8%) and C++ = 48 (23,1%)). This is a increase from [Holgeid 99] where 17,5% of the organizations used C++ and only 5,8% used Java. We can also see that COBOL is no longer as widespread as it was earlier. If we compare to [Holgeid 99] 10,7% of the organizations

[1] The average number of main systems which depend on data from other systems (2,3) divided with the average number of main systems in an organization (4,5).

used it, and there was made 114 systems with this language. In our questionnaire there is only 1,4% (1) of the organizations using COBOL, and only 1 system.

N	Min	Max	Sum	Mean	Median	Mode	SD
44	0	40	244	5,5	3	3	6,8

Table 5-30: Number of database systems used

Table 5-30 shows us that there are in average 5,5 database systems in every organization. This is in average 6,9 database systems less than in [Holgeid 99]. But there is still more database systems in average per organization than main systems, just as in [Holgeid 99]. Totally there are 244 database systems, divided on 44 organizations. Number of database systems varies from 0 to 40 per organization.

Database	N	Number org.	%	N	Number databases	%
Hierarkisk	44	d	6,1	44	11	4,5
Nettverk	44	10	15,2	44	26	10,7
Relasjon	44	36	54,5	44	129	52,9
Objektorientert	44	9	13,6	44	22	9,0
Annet	44	7	10,6	44	56	23,0

Table 5-31: Use of the databases

In Table 5-31 we see that 54,5% (36) of the organizations use relational data bases. 52,9% of all database systems used by the respondents are relational data bases. We can also see that network data bases are less used in our questionnaire (10,7% of all database systems used) than reported in [Holgeid 99] where 40,6% of all database systems used were network data bases.

5.6 Replacement systems

With replacement systems we mean systems that are developed to replace other systems or parts of them. Systems that mainly cover functionalities that already are covered by existing systems.

In this chapter we will look at some data with the replacement systems made in our respondent's organizations.

N	Min	Max	Sum	Mean	Median	Mode	SD
41	0	4	40	1,0	1	0	1,0

Table 5-32: Development of new systems

There are in average 1,0 new systems under development in each organization (Table 5-32). This is a decrease compared to [Krogstie 94] (2) and [Holgeid 99] (1,6). The number of systems under development spreads between 0 and 4 per organization. In our questionnaire the relation between the number of new systems under development and the total number main systems per organization 4,5/1 (22,2%) compared to 16,7% in [Holgeid 99], 23,7% in [Krogstie 94] and 17,6% in [Swanson and Beath 89].

N	Min	Max	Sum	Mean	Median	Mode	SD
41	0	3	23	,6	0	0	,8

Table 5-33: Share of replacement systems from number of new systems

From among the new systems, an average of 0,6 are replacement systems (Table 5-33). This means than 60% of all new developed systems are replacement systems, compared to 56% in [Holgeid 99] and 48% in [Krogstie 94].

Years	N	Min	Max	Sum	%	Mean	Median	Mode	SD
0-1 år	40	0	1	1	4,5	0,025	0	0	0,2
1-3 år	40	0	2	5	22,7	0,125	0	0	0,4
3-6 år	40	0	3	11	50,0	0,275	0	0	0,7
6-10 år	40	0	1	2	9,1	0,05	0	0	0,2
Mer enn 10 år	40	0	2	3	13,6	0,075	0	0	0,3

Table 5-34: Age distribution for the systems to be replaced

From Table 5-34 we can read that 50% (11) of the main systems, that are to be replaced, are between 3 and 6 years old. More than 70% of the main systems which will be replaced are older than 3 years. Only 4,5% are less than a year old. In [Holgeid 99] 34,1% were between 3 and 6 years, and 0 systems less than a year old.

	Our		[Holgeid 99]		[Krogsie 94]
Replacement reasons	**N**	**Mean**	**N**	**Mean**	**Mean**
Integrering med andre system	48	3,4	30	3,2	3,9
Standardisering	48	3,3	30	3,4	3,0
Ny teknisk arkitektur	48	3,0	30	2,9	3,7
Vanskelig å vedlikeholde	48	2,9	30	3,1	3,7
Finnes pakkeløsning	48	2,8	30	2,1	2,4
Vanskelig å drifte	48	2,6	30	2,3	3,7
Vanskelig å bruke	48	2,6	30	2,1	3,0
Finnes applikasjonsgenerator	48	1,9	30	1,6	1,8
Annen erstatningsgrunn	10	2,2			

Table 5-35: Reasons for replacement systems

In Table 5-35 we present different reasons to why organizations are making replacement systems. In our questionnaire the most important reasons for the replacement systems were “Integrering med andre systemer”, “Standarisering” og “Ny teknisk arkitektur”. In [Holgeid 99] “Standarisering”, “Integrering med andre systemer” and ”Vanskelig å vedlikeholde” were some of the main reasons. As we can see, the reasons for the replacement systems are almost the same today as 5 years ago.

Category	Frequency	Percent
Ingenting	6	14,0
Lite	9	20,9
En del	15	34,9
Mer enn halvparten	11	25,6
Mye	2	4,7
Total	**43**	**100,0**

Table 5-36: Reuse of programming-code in replacement-systems and new systems with overlapping functionality with existing systems

Because of the general opinion within IT that reuse of programming-code and also specifications and design, is one of the most efficient ways to increase the productivity in system-development, we found it interesting to examine to which degree organizations actually perform reuse, especially in connection to replacement systems.

As we see in Table 5-36, over 65% of the organizations do reuse programming-code “En del” or more. This is a big change compared to [Holgeid 99], where over 74% reused “Lite” or “Ingenting”. If we compare these results to [Krogstie 94] (86%), the change is even bigger.

Category	Frequency	Percent
Ingenting	17	41,5
Lite	10	24,4
En del	9	22,0
Mer enn halvparten	5	12,2
Mye	0	0,0
Total	**41**	**100,0**

Table 5-37: Reuse of specifications and design in replacement-systems and new systems with overlapping functionality with existing systems

From Table 5-37 we can see that only 34,2% of the organizations reuse specifications and design “En del” or more. The rest 65,9% reuse “Lite” or “Ingenting”. In [Holgeid 99] the situation was reported to be that 53,1% did reuse “Lite” or “Ingenting”. This means that there appear to be less specification and design reuse now than 5 years ago.

5.7 Organizing controls

With “organizing controls” we mean different procedures and functions which are used to control different aspects of maintenance.
In this chapter we will take a closer look at which organizing controls the organizations use when maintaining systems. The use of organizing controls has also been reported in earlier reports, [Lientz and Swanson 80], [Swanson and Beath 89], [Nosek and Palvia 90], [Henne 92], [Krogstie 94] and [Holgeid 99].

Organization controls	N	Number	%
Alle endringer i IT-systemene testes	47	35	74,5
Endringsforslag klassifiseres etter type og viktighet	47	30	63,8
Endringer av programvare blir dokumentert	47	27	57,4
Endringsforslag gjennomgår konsekvensanalyse og kostnadsestimering	47	26	55,3
Brukere som etterspør endringer får tilbakemelding uansett	47	24	51,1
Brukerkrav dokumenteres	47	23	48,9
Samme rutine for endringsforslag fra IT-avdeling og brukergrupper	47	19	40,4
Periodiske formelle gjennomganger av systemene	47	18	38,3
Ved akseptansetest av endringer oppdateres dokumentasjonene	47	16	34,0
Personellkostnader forbundet med drift og vedlikehold belastes brukergruppene	47	9	19,1
Utstyrskostnader forbundet med drift og vedlikehold belastes brukergruppene	47	8	17,0
Endringer blir samlet opp for periodisk implementasjon	47	6	12,8

Table 5-38: Use of organization controls in maintenance (sorted descending)

Table 5-38 shows us to what degree the different organizing controls are used in the respondent organizations. We can see that 74,5% of the organizations test all changes in the systems. This is an increase of 16% compared to [Holgeid 99] 58,5%. Also “Endringsforslag klasifiseres etter type og viktighet” (63,8%) has increased compared to [Holgeid 99] (58,5%). In [Holgeid 99] the three most used organizing controls where: “Brukerkrav dokumenteres”, “Endringsforslag klassifiseres etter type og viktighet” and “Alle endringer i IT-systemene testes”. This is very close to our report, which says that ”Alle endringer i IT-systemene testes”, “Endringsforslag klasifiseres etter type og viktighet” and “Endringer av programvare blir dokumentert” are the three most used organizing controls.

55,3% of the organizations in our report use “Endringsforslag gjennomgår konsekvensanalyse og kostnadsestimering” as a organization control. This is an increase from both [Krogstie 94] (54%) and [Holgeid 99] (36%).

5.8 Methods

This chapter will cover different descriptive data concerning the respondent organizations use of methods in system-development and maintenance.

There seems to be a general view that development and maintenance are influenced by the methods that are used. However, [Dekleva 92] reports that organizations that use modern development-methods do not seem to spend less time on maintenance than the other organizations. There is also reported that use of modern development-methods involved decreased number of errors in systems put in production, thereby also reduced amount of time used to correct production-errors. Further he reported that systems developed with a modern development-method were easier to change functionally then other systems. [Dekleva 92] concludes that although modern development-methods do not reduce the maintenance-share, they lead to increased service for the end-users, because their request for functional-changes can be carried out easier (i.e. increased functional development).

We will now look at aspects with documentation in maintenance-work, followed by different methods for resource-estimation and use of pre-defined methods in the different life cycles of the systems. We will end this chapter with different aspects with system development tools.

Documentation	Yes		No		Not available	
	N	%	N	%	N	%
Systemdokumentasjon	41	87,2	2	4,3	4	8,5
Testdokumentasjon	28	59,6	9	19,1	10	21,3
Brukerdokumentasjon	34	72,3	9	1,1	4	8,5
Programeringsspråkmanualer	23	48,9	11	23,4	13	27,7

Table 5-39: Use of documentation in maintenance-work

Table 5-39 shows us to what degree documentation is used in maintenance-work. 87,2% (41) of the organizations use system-documentation in maintenance-work, while 72,3% (34) use user-documentation. Both this numbers are a little higher then reported in [Holgeid 99] (78% (39) and 67,3% (35)). Test-documentation is used in 59,6% (28) of the organizations, which is considerably higher than in [Holgeid 99] (36% (18)). Programming-documentation is in our report used in 48,9% (23) of the organizations, which is lower then reported in [Holgeid 99] (59,2% (29)).

Documentation	N	Mean	Median	Mode
Systemdokumentasjonen	47	3,6	4	4
Testdokumentasjonen	47	3,0	3	3
Brukerdokumentasjonen	47	3,7	4	3

Table 5-40: Degree of up to date documentation

System- and user-documentation are in average for “Mer enn halvparten” of the systems up to date (Table 3-40). The test-documentations are “Middels” up to date. This reflects the results from Table 3-39. The documentations most used in maintenance-work, are also the ones that are most up to date.

Phase	Yes		No		Total
	N	%	N	%	N
Planlegging	20	43,5	26	56,5	46
Analyse	11	23,9	35	76,1	46
Utarbeidelse av kravspesifikasjon	26	56,5	20	43,5	46
Design	21	45,7	25	54,3	46
Implementasjon/programmering	24	52,2	22	47,8	46
Testing	25	54,3	21	45,7	46
Utrulling	15	32,6	31	67,4	46
Drift	17	37,0	29	63,0	46
Vedlikehold	13	28,3	33	71,7	46
Prosjektledelse	16	34,8	30	65,2	46

Table 5-41: Use of pre-defined methods in the life cycle of the systems

Table 5-42 shows the use of pre-defined methods in life cycles of the systems. Just over half of the organizations (56,5%) use a pre-define method for “Utarbeidelse av kravspesifikasjon”, “Testing” (54,3%) and “Implemenatasjon/programmering” (52,2%), while about 40% use a pre-defined method for “Design” (45,7%) and “Planlegging” (43,5%). Over 70% of the organizations do not use a pre-define method for “Vedlikehold”. Compared to [Holgeid 99] the number of organizations that do not use a pre-defined method for “Vedlikehold” is almost the same as in our report.

We will now look closer at different aspects of system development tools. This will in this report include all forms of automated-tools for development and maintenance of information-systems.

Yes		No		Total
N	%	N	%	N
22	53,7	19	46,3	41

Table 5-42: System development tools in development

53,7% (22) of the organizations use system development tools in development of new systems (Table 5-42). This is an increase compared to [Holgeid 99] (13,2%) and [Krogstie 94] (27,1%). The organizations that use system development tools in development have in average 137 employees, while organizations that do not have 75. It would appear that system development tools are most used in larger organizations. This was also reported in [Holgeid 99] and [Krogstie 94].

Yes		No		Total
N	%	N	%	N
16	39,0	25	61,0	41

Table 5-43: System development tools in maintenance

In maintenance 39% (16) of the organizations use system development tools (Table 5-43), compared to 11,3% in [Holgeid 99] and 10,6% in [Krogstie 94]. All organizations, but one, that used system development tools for maintenance also used it for development.

The percentage of organizations that use system development tools is increased compared to earlier reports, and we will take a closer look at how the organizations use their system development tools.

Phase	Yes		No		Total
	N	%	N	%	N
Planlegging	15	65,2	8	34,8	23
Analyse	9	39,1	14	60,9	23
Utarbeidelse av kravspesifikasjon	17	73,9	6	26,1	23
Design	15	65,2	8	34,8	23
Implementasjon/programmering	15	65,2	8	34,8	23
Testing	16	69,6	7	30,4	23
Utrulling	10	43,5	13	56,5	23
Drift	11	47,8	12	52,2	23
Vedlikehold	7	30,4	16	69,6	23
Prosjektledelse	12	52,2	11	47,8	23

Table 5-44: Use of system development tools

We can see that 73,9% of all the organizations that use system development tools use this technology in “Utarbeidelse av kravspesifikasjon”,69,6% in “Testing”, and 65,2% “Planlegging”, “Design” and “Implementasjon/programmering”. Even though 16 of 23 organizations use system development tools for maintenance, only 30,4% answer that they use system development tools in the life cycle “Vedlikehold”. We have not studied the reason for this, but one explanation can be that the organizations do not use a system development tool especially made for maintenance, and therefore did not answer YES on that current question.

N	Min	Max	Sum	Mean	Median	Mode	SD
23	2	10	96,5	4,2	4	3	1,8

Table 5-45: Number of years with system development tools experience

From Table 5-45 we can see that the organizations have in average just over 4 years of experience with system development tools. There are 2 organizations that have 2 years of experience, and one with 10 years of experience. This means that all the organizations that use system development tools have been using it for more than 2 years. This is a large increase compared to both [Holgeid 99] (57%) and [Krogstie 94] (42%).

N	Min	Max	Sum	Mean	Median	Mode	SD
23	0	6	44	1,9	1	1	1,6

Table 5-46: Number of existing main-systems that are supported by system development tools

There are in average just under 2 main-systems in each organization (organizations that use system development tools) that are supported by system development tools (Table 5-46). There are 3 organizations that do not have any main-systems that are supported by system development tools.

The use of system development tools is by many associated with increased efficiency in maintenance. [House 93] reported that in the short run productivity seems to decrease, while the quality of the IT-systems seems to increase.

5.9 Problem areas within maintenance

This chapter will look at the descriptive data on situation that the respondents experience more or less problematic in connection with IT-system maintenance.

Problem areas	5 i %	4 i %	3 i %	2 i %	1 i %	N	Mean
IT-systemets orginalkvalitet	17,0	38,3	31,9	8,5	4,3	47	3,6
Trange budsjetter	23,4	34,0	23,4	8,5	10,6	47	3,5
Kvalitet på systemdokumentasjonen	10,6	31,9	36,2	19,1	2,1	47	3,3
Tilgjengelighet av personell	4,3	29,8	42,6	19,1	4,3	47	3,1
Utskifting av personell	4,3	36,2	36,2	12,8	10,6	47	3,1
Endring av teknisk arkitektur	8,5	19,1	38,3	29,8	4,3	47	3,0
Urealistiske brukerforventninger	6,4	31,9	19,1	36,2	6,4	47	3,0
Utvidede brukerkrav	4,3	25,5	38,3	19,1	12,8	47	2,9
Manglende brukerforståelse av systemet	10,6	19,1	27,7	31,9	10,6	47	2,9
Vedlikeholdspersonellets dyktighet	6,4	14,9	46,8	27,7	4,3	47	2,9
Mangelfull opplæring av brukere	4,3	19,1	40,4	27,7	8,5	47	2,8
Driftsfeil	17,0	10,6	21,3	36,2	14,9	47	2,8
Intern konkurranse om personell	6,4	19,1	34,0	29,8	10,6	47	2,8
Pålitelighet til teknisk arkitektur	8,5	17,0	25,5	36,2	12,8	47	2,7
Maskinhastighet	4,3	19,1	25,5	31,9	19,1	47	2,6
Dataintegritet i applikasjonen	8,5	14,9	27,7	29,8	19,1	47	2,6
Manglende brukerinteresse	6,4	12,8	25,5	34,0	21,3	47	2,5
Ikke bruk av standarder	6,4	12,8	27,7	34,0	19,1	47	2,5
Personellets motivasjon	0,0	10,6	36,2	34,0	19,1	47	2,4
Personellets produktivitet	2,1	12,8	29,8	36,2	19,1	47	2,4
Manglende støtte av ledelsen	12,8	6,4	19,1	21,3	40,4	47	2,3
Utskifninger i brukerorganisasjonen	2,1	12,8	23,4	38,3	23,4	47	2,3
Datalagringskrav	4,3	10,6	19,1	31,9	34,0	47	2,2
Annet	0,0	0,0	0,0	20,0	80,0	5	1,2

Table 5-47: Problem areas within maintenance (sorted decreasing on average)

In Table 5-47 we can see different problem areas within maintenance of IT-systems, sorted decreasing on average degree of importance. We can see that "IT-systemets orginalkvalitet" and "Trange budsjetter" are in average considered as "Større problem", while "Manglende støtte av ledelsen", "Utskiftninger I brukerorganisasjonen" and "Datalagringskrav" are in average considered as "Liten grad et problem". [Lientz and Swanson 80] reported that user knowledge, programmer's efficiency and product-quality was important, while [Nosek and Palvia 90] found that the programmers availability and efficiency were among the most important problem areas. [Krogstie 94] reported "Utskiftning av personell" and "Kvalitet på systemdokumentasjonene" as important problem areas, while [Holgeid 99] reported "Kvalitet på dokumentasjonen", "Utskifting av personell" and "personellets tilgjengelighet" as the most important problems.

6 Hypothesis-testing

In this chapter we will present the results from hypothesis-testing. In chapter 6.1 we will look at normality testing of the maintenance variables, in chapter 6.2 we will present the testing of the hypothesis in the "maintenance/development" group, in chapter 6.3 we will look at the "size and type of organizations", in chapter 6.4 "importance of IT", in chapter 6.5 "internal competence", chapter 6.6 "complexity of the portfolio", chapter 6.7 "organizing", chapter 6.8 "use of methods" and in chapter 6.9 we will look at hypothesis within "use of tools".

6.1 Normality test of the maintenance variables

While testing our hypothesis we will, as mentioned in chapter 3, use different statistical techniques depending on the result of the variables normality tests.

Category	Skewness	Kurtosis	Shapiro-Wilks	Sign.	Kolmogorov-Smirnov	Lillefors Sign.
Total andel vedlikehold[2]	0,392	-0,087	0,966	0,407	0,153	0,055
Vedlikehold hovedsakelig[3]	0,222	-0,120	0,966	0,407	0,153	0,055
Prosent vedlikehold[4]	-0,074	-0,562	0,990	0,988	0,066	0,200
Prosent funksjonell vedlikehold[5]	0,238	-0,326	0,971	0,522	0,092	0,200

Table 6-1: Normality test of the maintenance variables

Table 6-1 shows us different normality test of the maintenance variables. We can not reject the hypothesis that these are normally distributed. The variables are not completely normally distributed, since skewness and kurtosis are not nil, but the absolute value is less than 1.

6.2 Maintenance/development

H1: There is no difference in the amount of work-load used on maintenance and development, when we only look at maintenance and development.

Prosent vedlikehold			Prosent utvikling				
N	**Mean**	**SD**	**N**	**Mean**	**SD**	**Δ**	**P**
52	65,9	21,4	52	34,1	21,4	31,8	**<0,0005**

Table 6-2: Maintenance vs. development when we look away from other work tasks

H1 is rejected; from Table 6-2 we can see that there is significantly more maintenance than development when we look away from other work tasks.

---

[2] Total share traditional maintenance.
[3] Total share traditional maintenance in organizations where the IT-departments work mostly with development and maintenance (>50% of the work is development and maintenance).
[4] Total share traditional maintenance when we only look at development- and maintenance-work.
[5] Total share functional maintenance when we only look at functional development and functional maintenance.

H2: There is no difference in the amount of work-load used on maintenance and development.

Total andel vedlikehold			Total andel utvikling				
N	**Mean**	**SD**	**N**	**Mean**	**SD**	**Δ**	**p**
52	35,9	15,6	52	21,9	17,8	14,0	**<0,0005**

Table 6-3: Maintenance vs. development

From Table 6-3, we can see that there is still significantly more maintenance than development amongst our respondents, even when we consider other form for work beside maintenance and development. So we reject H2. Also [Krogstie 94] and [Holgeid 99] rejected this hypothesis. Both [Krogstie 94] and [Holgeid 99] had a larger average difference between share maintenance and share development than in our investigation. 14% in our investigation compared to 17% in [Krogstie 94] and 24% [Holgeid 99].

H3: There is no difference between the work-load used on functional maintenance and traditional maintenance, when we look at development and maintenance only.

Prosent funksjonell vedlikehold			Prosent vedlikehold				
N	**Mean**	**SD**	**N**	**Mean**	**SD**	**Δ**	**p**
52	61,1	20,2	52	65,9	21,4	-4,8	0,188

Table 6-4: Functional maintenance vs. traditional maintenance when we look away from other work tasks

We can see from Table 6-4 that there is no significant difference (p=0,188) between functional maintenance and traditional maintenance when we look at development and maintenance only. From this result we do not reject H3.

Both [Krogstie 94] (p<0,0005) and [Holgeid 99] (p=0,015) rejected this hypothesis.

H4: There is no difference between the work-load used on functional development and traditional development, when we look at development and maintenance only.

Prosent funksjonell utvikling			Prosent utvikling				
N	**Mean**	**SD**	**N**	**Mean**	**SD**	**Δ**	**P**
52	38,9	20,2	52	34,1	21,4	4,8	0,188

Table 6-5: Functional development vs. traditional development when we look away from other work tasks

As H3, H4 is not rejected. Functional development is not significantly larger than traditional development.

H5: There is no difference between the work-load used on functional development and functional maintenance, when we look at development and maintenance only.

Prosent funksjonell utvikling			Prosent funksjonell vedlikehold				
N	**Mean**	**SD**	**N**	**Mean**	**SD**	**Δ**	**p**
52	38,9	20,2	52	61,1	20,2	-22,2	**0,001**

Table 6-6: Functional development vs. functional maintenance

From Table 6-6 we can see that functional maintenance is significantly larger compared to functional development. Thereby we reject H5. [Krogstie 94] reported that functional development was significantly larger than functional maintenance, while [Holgeid 99] concluded similar to us. In our investigation functional maintenance amounts to 61,1%, while in [Holgeid 99] it was 62,3% when we look at development and maintenance only.

6.3 Size and type of organization

H6: There is no difference in the amount of maintenance-work between organizations with many employees and organizations with fewer employees.

	>=27			<27				
	N	**Mean**	**SD**	**N**	**Mean**	**SD**	**Δ**	**p**
Total andel vedlikehold	27	33,9	13,9	25	38,0	17,4	-4,1	0,348
Vedlikehold hovedsakelig	16	38,7	14,5	16	44,6	15,8	-5,9	0,277
Prosent vedlikehold	27	64,6	19,5	25	67,2	23,6	-2,6	0,663
Prosent funksjonell vedlikehold	27	61,7	15,8	25	60,3	24,3	1,4	0,807

Table 6-7: Maintenance vs. number of employees

Table 6-7 shows us share maintenance vs. total number of employees, divided into 2 categories sat by median-value of total number employees (27). With a significance-level at 5% we can see that we can not reject H6 for any of the maintenance-types. [Krogstie 94] did not reject this hypothesis either, while [Holgeid 99] rejected it for “Vedlikehold hovedsakelig” and “Prosent vedlikehold”.

	Antall ansatte		
	S	**N**	**p**
Total andel vedlikehold	-0,210	52	0,135
Vedlikehold hovedsakelig	-0,221	32	0,225
Prosent vedlikehold	0,020	52	0,888
Prosent funksjonell vedlikehold	0,200	52	0,155

Table 6-8: Correlation, maintenance and number of employees

From the correlation-analyses in Table 6-8, we can see that there are no significant linear-relation between any of the maintenance-types and number of total employees. [Holgeid 99] found significance and linear-relation between “Vedlikehold hovedsakelig” (p=0,012) and “Prosent funksjonell vedlikehold” (p=0,012) and number of total employees.

6.4 Importance of IT

H7: There is no difference in the amount of maintenance-work between organizations where IT is of big strategic importance and organizations where IT is of less strategic importance.

	Absolutt strategisk betydning			Stor/tildels/lite/ikke strategisk betydning				
	N	Mean	SD	N	Mean	SD	Δ	p
Total andel vedlikehold	41	36,5	15,1	11	33,6	18,1	2,9	0,593
Vedlikehold hovedsakelig	27	40,6	1595	5	47,4	10,3	-6,8	0,367
Prosent vedlikehold	41	64,0	21,4	11	73,0	20,9	-9,0	0,217
Prosent funksjonell vedlikehold	41	61,3	20,2	11	60,2	21,0	1,1	0,883

Table 6-9: Maintenance vs. strategic importance of IT in the organization

In Table 6-9 we can see that there are no significant differences between the different maintenance-types and how strategic IT is for the organizations. In chapter 5.2 we could see that IT, for 79,6% of the respondent organizations, was of “Absolutt” strategic importance. We will not reject H7.

H8: There is no difference in the amount of maintenance-work between organizations in which the size of the IT-department compared to the total number of employees is large and the organizations where the size of the IT-department compared to the total number of employees is small.

	>=0,09			<0,09				
	N	Mean	SD	N	Mean	SD	Δ	p
Total andel vedlikehold	25	39,6	17,5	27	32,4	13,1	7,2	0,098
Vedlikehold hovedsakelig	20	42,3	18,0	12	40,7	9,6	1,6	0,781
Prosent vedlikehold	25	62,0	24,8	27	69,5	17,4	-7,5	0,209
Prosent funksjonell vedlikehold	25	58,6	22,7	27	63,3	17,6	-4,7	0,402

Table 6-10: Maintenance vs. number of employees in the IT-department in proportion to total number of employees

In Table 6-10 the maintenance-types are divided by the median-value of number of employees in the IT-department / Total number of employees. We can see that H8 is not rejected for any of the maintenance-types, but we can still see that “Total andel vedlikehold” is a little smaller for the organizations with few employees in the IT-department in proportion to total number of employees (p=0,098). [Holgeid 99] rejected the hypothesis for “Prosent funksjonell vedlikehold”, while [Krogstie 94] did not reject the hypothesis for any of this maintenance-types, just like us.

	Data employees/ total employees		
	S	**N**	**p**
Total andel vedlikehold	0,172	52	0,223
Vedlikehold hovedsakelig	-0,108	32	0,556
Prosent vedlikehold	-0,164	52	0,246
Prosent funksjonell vedlikehold	-0,250	52	0,073

Table 6-11: Correlation, maintenance and number of employees in the IT-department in proportion to total number of employees

There is no significance or linear-relation between maintenance-types and the number of employees in the IT-department in proportion to total number of employees (Table 6-11).

H9: There is no difference in the amount of maintenance-work between organizations in which there are many system-developers in proportion to total number of end-users, and organizations with few system-developers in proportion to total number of end-users.

	>=0,02			<0,02				
	N	**Mean**	**SD**	**N**	**Mean**	**SD**	**Δ**	**P**
Total andel vedlikehold	25	38,1	15,7	25	33,8	15,6	4,3	0,330
Vedlikehold hovedsakelig	17	42,8	15,5	15	40,3	15,3	2,5	0,652
Prosent vedlikehold	25	64,6	22,2	25	65,7	20,7	-1,1	0,861
Prosent funksjonell vedlikehold	25	57,2	20,0	25	63,0	19,5	-5,8	0,314

Table 6-12: Maintenance vs. number of system-developers in proportion to total number of end-users

This is a similar test to the one in H8, but here we only look at the number of developers in the IT-department in proportion to total number of end-users, and use the median-value as a cut-point. Since there is no significant results (Table 6-12), we can not reject H9.

[Holgeid 99] rejected this hypothesis for “Total andel vedlikehold” (p=0,026) and “Vedlikehold hovedsakelig” (p=0,047). [Krogstie 94] did not reject this hypothesis for any of the maintenance-types.

	System-developers/ end-users		
	S	**N**	**p**
Total andel vedlikehold	0,171	50	0,236
Vedlikehold hovedsakelig	0,074	32	0,689
Prosent vedlikehold	-0,022	50	0,878
Prosent funksjonell vedlikehold	-0,173	50	0,231

Table 6-13: Correlation, maintenance and number of system-developers in proportion to total number of end-users

From the correlation-analyses in Table 6-13 we can not see any significance or linear-relation between the maintenance-types and system-developers in proportion to end-users.

H10: There is no difference in the amount of maintenance-work between organizations with large IT-budgets in proportion to total number of employees, and organizations with small IT-budgets in proportion to total number of employees.

	>=0,24			<0,24				
	N	**Mean**	**SD**	**N**	**Mean**	**SD**	**Δ**	**p**
Total andel vedlikehold	25	38,0	17,4	27	33,9	13,9	4,1	0,348
Vedlikehold hovedsakelig	16	44,6	15,8	16	38,7	14,5	5,9	0,277
Prosent vedlikehold	25	67,2	23,6	27	64,6	19,5	2,6	0,663
Prosent funksjonell vedlikehold	25	60,3	24,3	27	61,7	15,8	-1,4	0,807

Table 6-14: Maintenance vs. IT-budget for 2003 (mill NOK) in proportion to total number of employees

In Table 6-14 we have compared the different maintenance-types and the IT-budget for 2003 in proportion to total number of employees, where we used the median-value as a cut-point. There is no statistical significant difference in the amount of maintenance-work performed between organizations with larger IT-budget in proportion to the total number of employees compared to organizations with smaller IT-budget in proportion to the total number of employees; we can therefore not reject H10 for any of the maintenance-types.

6.5 Internal competency

H11: There is no difference in the amount of maintenance-work between organizations with system-developers who have in average many years of experience in the IT-department, and organizations with system-developers who have shorter experience.

	>=4,50			<4,50				
	N	**Mean**	**SD**	**N**	**Mean**	**SD**	**Δ**	**p**
Total andel vedlikehold	22	34,5	14,7	17	37,9	18,5	-3,4	0,522
Vedlikehold hovedsakelig	15	38,7	14,3	11	45,9	17,1	-7,2	0,256
Prosent vedlikehold	22	62,2	24,2	17	68,5	20,7	-6,3	0,398
Prosent funksjonell vedlikehold	22	58,1	19,6	17	60,5	21,3	-2,4	0,716

Table 6-15: Maintenance vs. system-developers average IT-department experience

There is no significant difference in maintenance between organization with system-developers with long average IT-department experience and organizations with system-developers with short average IT-department experience (Table 6-15). Long/short average It-department experience is divided by the median-value. H11 is therefore not rejected.

	System-developers average IT-department experience		
	S	**N**	**p**
Total andel vedlikehold	-0,081	39	0,626
Vedlikehold hovedsakelig	-0,330	26	0,100
Prosent vedlikehold	-0,142	39	0,388
Prosent funksjonell vedlikehold	-0,050	39	0,764

Table 6-16: Correlation, maintenance and system-developers average IT-department experience

From the correlation-analysis in Table 6-16, we can not see any significance or linear-relation between the maintenance-types and system-developers average IT-department experience

H12: There is no difference in the amount of maintenance-work between organizations with system-developers who have many years of experience in the IT-department in proportion to the main-systems average age, and organizations with system-developers who have short experience in proportion to the average age of the main-systems.

	>=1,33			<1,33				
	N	**Mean**	**SD**	**N**	**Mean**	**SD**	**Δ**	**p**
Total andel vedlikehold	19	37,4	15,4	18	34,6	17,7	2,8	0,610
Vedlikehold hovedsakelig	14	38,5	17,3	11	45,2	13,8	-6,7	0,306
Prosent vedlikehold	19	62,4	26,8	18	68,1	19,2	-5,7	0,465
Prosent funksjonell vedlikehold	19	53,5	21,3	18	65,8	17,8	-12,3	0,065

Table 6-17: Maintenance vs. the system-developers experience in the IT-department in proportion to the main-systems average age

In Table 6-17 we do not see any significant difference in maintenance-work in organizations where the proportion between system-developers experience in the IT-department and the main-systems average age is small and organizations where this proportion is large. Small/large is decided by the median-value. We can therefore not reject H12, although large difference relative to functional maintenance.

We can see a difference in “Prosent funksjonell vedlikehold”, but as said earlier, it is not significant.

H13: There is no difference in the amount of maintenance-work between organizations with many hired IT-consultants within system-developing per system-developer, and organizations with few hired IT-consultants within system-developing per system-developer.

	>=0,04			<0,04				
	N	Mean	SD	N	Mean	SD	Δ	p
Total andel vedlikehold	20	36,6	17,0	18	37,0	14,7	-0,4	0,947
Vedlikehold hovedsakelig	14	40,4	18,4	12	43,3	12,3	-2,9	0,647
Prosent vedlikehold	20	63,0	24,7	18	65,2	19,8	-2,2	0,758
Prosent funksjonell vedlikehold	20	54,5	17,0	18	62,0	21,3	-7,5	0,238

Table 6-18: Maintenance vs. number of hired IT-consultants in proportion to number of system-developers

Table 6-18 shows us maintenance vs. number of hired IT-consultants in proportion to number of system-developers, where the cut-point is sat by the median-value. From the table we can see that there is no significant difference for any of the maintenance-types, and therefore we can not reject H13 for any of them. [Holgeid 99] rejected this hypothesis for all the maintenance-types except for "Prosent funksjonell vedlikehold".

	Hired IT-consultants/ system developers		
	S	N	p
Total andel vedlikehold	0,012	38	0,944
Vedlikehold hovedsakelig	-0,060	26	0,769
Prosent vedlikehold	0,086	38	0,608
Prosent funksjonell vedlikehold	0,007	38	0,965

Table 6-19: Correlation, maintenance and number of hired IT-consultants in proportion to number of system-developers

The correlation-analyses in Table 6-19 shows no significance or linear-relation between the different maintenance-types and the proportion between hired IT-consultants and number of system-developers.

6.6 Complexity of the portfolio

H14: There is no difference in the amount of maintenance-work between organizations with many main-systems and organizations with fewer main-systems.

	>=3,00			<3,00				
	N	Mean	SD	N	Mean	SD	Δ	p
Total andel vedlikehold	38	34,9	13,1	14	38,6	21,4	-3,7	0,461
Vedlikehold hovedsakelig	25	38,7	13,5	7	52,1	17,5	-13,4	**0,037**
Prosent vedlikehold	38	62,9	20,6	14	74,0	22,3	-11,1	0,098
Prosent funksjonell vedlikehold	38	61,0	17,3	14	61,1	27,3	-0,1	0,993

Table 6-20: Maintenance vs. number of main-systems

In Table 6-20 we can see the comparison between the different maintenance-types and number of main-systems. The cut-point is decided by the median-value. We find a significant difference between organizations with few main-systems and organizations with many main-systems for "Vedlikehold hovedsakelig" (p=0,037). We will therefore reject H14 for "Vedlikehold hovedsakelig" but not for the other maintenance-types. Both

[Holgeid 99] and [Krogstie 94] did not find any significant difference for any of this maintenance-types.

H15: There is no difference in the amount of maintenance-work between organizations with many end-users and organizations with few end-users.

	>=55,00			<55,00				
	N	**Mean**	**SD**	**N**	**Mean**	**SD**	**Δ**	**p**
Total andel vedlikehold	26	36,2	17,8	25	36,3	13,4	-0,1	0,977
Vedlikehold hovedsakelig	18	41,4	18,6	14	42,0	10,2	-0,6	0,913
Prosent vedlikehold	26	61,2	22,2	25	70,7	20,3	-9,5	0,116
Prosent funksjonell vedlikehold	26	57,1	18,1	25	65,0	22,1	-7,9	0,167

Table 6-21: Maintenance vs. total number of end-users

In Table 6-21 we can see that there is no significant difference in maintenance between organizations with many total end-users and organizations with few total end-user, where many/few is decided by the median-value. We do not reject H15.

	Number of end-users		
	S	**N**	**p**
Total andel vedlikehold	-0,129	51	0,368
Vedlikehold hovedsakelig	-0,202	32	0,266
Prosent vedlikehold	-0,319	51	**0,022**
Prosent funksjonell vedlikehold	-0,267	51	0,058

Table 6-22: Correlation, maintenance and total number of end-users

From the correlation-analyses in Table 6-22 we can see that "Prosent vedlikehold" has a linear-relation with number of total end-users in such a way that fewer end-users involve an increase in the "Prosent vedlikehold".

	>=16,00			<16,00				
	N	**Mean**	**SD**	**N**	**Mean**	**SD**	**Δ**	**p**
Total andel vedlikehold	27	34,8	14,0	25	37,1	17,5	-2,3	0,610
Vedlikehold hovedsakelig	15	40,9	14,4	17	42,4	16,3	-1,5	0,786
Prosent vedlikehold	27	67,8	19,0	25	63,8	23,9	4,0	0,515
Prosent funksjonell vedlikehold	27	64,2	15,4	25	57,7	24,2	6,5	0,261

Table 6-23: Maintenance vs. number of internal end-users

We took also a look at the difference in maintenance vs. internal end-users. In Table 6-23 we can see that there is no significant difference in maintenance between organizations with few internal end-users and organizations with many internal end-users where many/few is decided by the median-value.

H16: There is no difference in the amount of maintenance-work between organizations with main-systems with high age average, and organizations with main-systems with low age average.

	>=2,42			<2,42				
	N	Mean	SD	N	Mean	SD	Δ	p
Total andel vedlikehold	25	35,1	15,7	24	36,2	16,0	-1,1	0,811
Vedlikehold hovedsakelig	14	42,6	16,1	17	40,4	15,2	2,2	0,705
Prosent vedlikehold	25	66,5	20,7	24	64,2	22,6	2,3	0,713
Prosent funksjonell vedlikehold	25	62,0	18,9	24	59,1	20,9	2,9	0,616

Table 6-24: Maintenance vs. the main-systems average age

In Table 6-24 we do not find any significant differences in maintenance between organizations with main-systems with a high average age and organizations that have main-systems with low average age. Therefore we can not reject H16.

H17: There is no difference in the amount of maintenance-work between organizations with main-systems which are highly dependent on data from other systems, and organizations with main-systems which are less dependent on data from other systems.

	>=2,00			<2,00				
	N	Mean	SD	N	Mean	SD	Δ	p
Total andel vedlikehold	25	34,4	14,0	23	35,8	16,5	-1,4	0,748
Vedlikehold hovedsakelig	17	38,7	13,4	13	42,2	16,3	-3,5	0,530
Prosent vedlikehold	25	60,5	21,0	23	70,5	22,4	-10,0	0,117
Prosent funksjonell vedlikehold	25	57,5	17,6	23	63,4	23,4	-5,9	0,324

Table 6-25: Maintenance vs. the main-systems dependency on data from other systems

There is no significant differences in maintenance between organizations that have main-systems that depend on data from many other systems and organizations who have main-systems that depend on data from few other systems, where many/few is decided by the median-value. H17 is therefore not rejected.

	Main-systems dependency on data from other systems		
	S	N	p
Total andel vedlikehold	-0,101	48	0,496
Vedlikehold hovedsakelig	-0,159	30	0,400
Prosent vedlikehold	-0,279	48	0,055
Prosent funksjonell vedlikehold	-0,106	48	0,475

Table 6-26: Correlation, maintenance and main-systems dependency on data from other systems

From Table 6-26 we can see that there is no correlation between these variables. We can see that “Prosent vedlikehold” has a small p-value (p=0,055), this is similar to [Holgeid 99] where “Prosent vedlikehold” correlated in such a way that “Prosent vedlikehold” decreased when main-systems dependency on data from other systems increased.

H18: There is no difference in the amount of maintenance-work between organizations with many different system-configurations and organizations with fewer different system-configurations.

	>=2,00			<2,00				
	N	**Mean**	**SD**	**N**	**Mean**	**SD**	**Δ**	**p**
Total andel vedlikehold	26	35,5	16,6	19	39,6	14,2	-4,1	0,382
Vedlikehold hovedsakelig	18	39,8	17,9	11	46,0	9,4	-6,2	0,302
Prosent vedlikehold	26	61,0	23,0	19	75,0	18,9	-14,0	**0,035**
Prosent funksjonell vedlikehold	26	59,3	18,8	19	64,8	22,7	-5,5	0,386

Table 6-27: Maintenance vs. number of different system-configurations

Organizations with few different system-configurations do significantly more maintenance than organizations with many different system-configurations, when we only look at maintenance and development. Few/many are here decided by the median-value of number of different system-configurations.

H19: There is no difference in the amount of maintenance-work between organizations that use many different programming-languages, and organizations that use fewer different programming-languages.

	>=2,00			<2,00				
	N	**Mean**	**SD**	**N**	**Mean**	**SD**	**Δ**	**p**
Total andel vedlikehold	18	29,6	11,7	17	44,9	15,0	-15,3	**0,002**
Vedlikehold hovedsakelig	13	31,8	12,5	11	50,7	13,0	-18,9	**0,001**
Prosent vedlikehold	18	51,2	19,6	17	74,7	20,0	-23,5	**0,001**
Prosent funksjonell vedlikehold	18	57,1	20,1	17	59,5	20,6	-2,4	0,735

Table 6-28: Maintenance vs. number of different programming-languages

There is significantly more “Total andel vedlikehold”, “Vedlikehold hovedsakelig” and “Prosent vedlikehold” in organizations with few different programming-languages than in organizations with many different programming-languages, where few/many is decided by the median-value of number of different programming-languages. H19 is therefore rejected for the named variables, but not for “Prosent funksjonell vedlikehold”.

	Number of different programming-languages		
	S	**N**	**p**
Total andel vedlikehold	-0,491	35	**0,003**
Vedlikehold hovedsakelig	-0,549	24	**0,005**
Prosent vedlikehold	-0,442	35	**0,008**
Prosent funksjonell vedlikehold	-0,105	35	0,550

Table 6-29: Correlation, maintenance and number of different programming-languages

The correlation-analyses in Table 6-29 verify the indication from Table 6-28: “Total andel vedlikehold”, “Vedlikehold hovedsakelig” and “Prosent vedlikehold” decreases when number of different programming-languages increase.

Intuitively we would think that maintenance increased when the complexity increased. Number of different programming-languages could be a factor of complexity. That the results seem to be the opposite can be explained by other factors influencing the results.

6.7 Organizing

H20: There is no difference in the amount of maintenance-work between organizations where the maintenance-workers are organized differently from the developers and organizations where there is no such difference.

	Lik			Forskjellige				
	N	**Mean**	**SD**	**N**	**Mean**	**SD**	**Δ**	**p**
Total andel vedlikehold	35	35,4	16,4	14	35,1	13,1	0,3	0,947
Vedlikehold hovedsakelig	24	40,7	15,6	7	41,4	12,5	-0,7	0,907
Prosent vedlikehold	35	65,0	23,1	14	65,1	18,6	-0,1	0,979
Prosent funksjonell vedlikehold	35	57,4	21,0	14	67,4	17,4	-10,0	0,123

Table 6-30: Maintenance vs. the organization of the maintenance-workers

In Table 6-30 we do not see any significance in maintenance between the organizations that organize the maintenance-workers differently from the developers and the organizations that do not organize them differently. H20 is therefore not rejected. In the sub-selection “Forskjellige” include both the organizations that formally organize the maintenance-workers differently, and the organizations that do this informally.

H21: There is no difference in the amount of maintenance-work between organizations where maintenance is often performed by the people who developed the system, and organizations where maintenance is rarely performed by the people who developed the system.

	Alltid/ofte			Av og til /skjelden/aldri				
	N	**Mean**	**SD**	**N**	**Mean**	**SD**	**Δ**	**p**
Total andel vedlikehold	24	37,2	16,9	25	34,3	15,4	2,9	0,528
Vedlikehold hovedsakelig	17	41,6	17,5	14	41,3	13,3	0,3	0,950
Prosent vedlikehold	24	63,1	21,3	25	67,7	22,7	-4,6	0,466
Prosent funksjonell vedlikehold	24	58,8	19,9	25	62,4	21,6	-3,6	0,544

Table 6-31: Maintenance vs. degree of how often the maintenance is performed by the same people that developed the system

We can not find any significant differences in maintenance between organizations where maintenance is “Alltid/ofte” performed by the same people that developed the systems and organizations where maintenance is “Av og til/skjelden/aldri” performed by the same people that developed the systems. H21 is therefore not rejected.

6.8 Use of methods

H22: There is no difference in the amount of maintenance-work between organizations that use methodologies in the IT-systems lifecycle, and the organizations that do not use this.

	Bruker en metode i livssyklusen			Bruker ikke en metode i livssyklusen				
	N	**Mean**	**SD**	**N**	**Mean**	**SD**	**Δ**	**p**
Total andel vedlikehold	36	37,0	15,1	8	33,2	17,4	3,8	0,541
Vedlikehold hovedsakelig	25	41,2	15,3	3	41,7	12,6	-0,5	0,964
Prosent vedlikehold	36	63,5	22,1	8	79,0	20,7	-15,5	0,076
Prosent funksjonell vedlikehold	36	56,0	19,6	8	79,5	18,5	-23,5	**0,004**

Table 6-32: Maintenance vs. the use of methodologies in the IT-systems lifecycle

From Table 6-32 we can see that there is a significant difference in “Prosent funksjonell vedlikehold” between the organizations that use methodologies in the IT-systems lifecycle and the organizations that do not use this. There is also a noticeable difference in “Prosent vedlikehold”, but not significant.

	Bruker en metode i analyse			Bruker ikke en metode i analyse				
	N	**Mean**	**SD**	**N**	**Mean**	**SD**	**Δ**	**p**
Total andel vedlikehold	11	25,3	7,6	33	40,0	15,7	-14,7	**0,005**
Vedlikehold hovedsakelig	8	26,3	8,2	20	47,3	12,5	-21,0	**0,000**
Prosent vedlikehold	11	48,8	23,9	33	72,2	18,9	-23,4	**0,002**
Prosent funksjonell vedlikehold	11	55,6	22,5	33	61,8	21,0	-6,2	0,408

Table 6-33: Maintenance vs. The use of methodologies in the analysis-phase

We will take a closer look at some of the different IT-systems lifecycle phases. In Table 6-33 we can see that there is significance in “Total andel vedlikehold”, “Vedlikehold hovedsakelig” and “Prosent vedlikehold” between the organizations that use methodologies in the analysis-phase and the organizations that do not use this.

	Bruker en metode i kravspek			Bruker ikke en metode i kravspek				
	N	**Mean**	**SD**	**N**	**Mean**	**SD**	**Δ**	**p**
Total andel vedlikehold	26	34,8	15,9	18	38,5	15,0	-3,7	0,442
Vedlikehold hovedsakelig	19	38,9	16,2	9	46,3	10,5	-7,4	0,222
Prosent vedlikehold	26	60,0	22,8	18	75,4	18,9	-15,4	**0,023**
Prosent funksjonell vedlikehold	26	57,2	22,2	18	64,7	19,7	-7,6	0,252

Table 6-34: Maintenance vs. The use of methodologies in the requirement-specification phase

In Table 6-34 we can see that organizations that use methodologies in the requirement-specification phase do significantly less traditional-maintenance than organizations that do not use methodologies in the requirement-specification phase.

	Bruker en metode i implementasjon			**Bruker ikke en metode i implementasjon**				
	N	**Mean**	**SD**	**N**	**Mean**	**SD**	**Δ**	**p**
Total andel vedlikehold	24	36,3	17,1	20	36,3	13,6	0,0	0,994
Vedlikehold hovedsakelig	16	41,8	17,5	12	40,6	11,1	1,2	0,833
Prosent vedlikehold	24	64,0	23,7	20	69,2	21,1	-5,2	0,448
Prosent funksjonell vedlikehold	24	54,4	21,4	20	67,4	19,4	-13,0	**0,043**

Table 6-35: Maintenance vs. The use of methodologies in the implementation-phase

We see from Table 6-35 that there is a significant difference in "Prosent funksjonell vedlikehold" between organizations that use methodologies in the implementation-phase and organizations that do not use this.

	Brukerkrav dokumenteres			**Brukerkrav dokumenteres ikke**				
	N	**Mean**	**SD**	**N**	**Mean**	**SD**	**Δ**	**p**
Total andel vedlikehold	22	35,8	13,7	23	36,1	17,3	-0,3	0,954
Vedlikehold hovedsakelig	16	40,0	12,9	12	43,1	17,5	-3,1	0,589
Prosent vedlikehold	22	57,6	18,5	23	73,7	23,4	-16,1	**0,015**
Prosent funksjonell vedlikehold	22	53,4	19,7	23	66,6	20,7	-13,2	**0,034**

Table 6-36: Maintenance vs. The documentation of user-demands

We tested if documentation has any affect on maintenance, and we found out (Table 6-36) that there is significant less "Prosent vedlikehold" and "Prosent funksjonell vedlikehold" in organizations that do document user-demands compared to organizations that do not.

	Endringer dokumenteres			**Endringer dokumenteres ikke**				
	N	**Mean**	**SD**	**N**	**Mean**	**SD**	**Δ**	**p**
Total andel vedlikehold	27	35,9	15,7	18	36,1	15,5	-0,2	0,972
Vedlikehold hovedsakelig	19	39,9	16,1	9	44,2	12,0	-4,3	0,482
Prosent vedlikehold	27	59,3	21,0	18	75,6	21,4	-16,3	**0,015**
Prosent funksjonell vedlikehold	27	52,2	18,7	18	72,2	19,0	-20,0	**0,001**

Table 6-37: Maintenance vs. The documentation of alteration

As we can see from Table 6-37 there is also a significant difference in "Prosent vedlikehold" and "Prosent funksjonell vedlikehold" between organizations that do document alterations and organizations that do not.

6.9 Use of tools

H23: There is no difference in the amount of maintenance-work between organizations that use system development tools, and the organizations that do not use this.

	Bruker CASE			Bruker ikke CASE				
	N	Mean	SD	N	Mean	SD	Δ	p
Total andel vedlikehold	21	36,4	16,3	18	37,2	13,6	-0,8	0,872
Vedlikehold hovedsakelig	17	38,3	17,4	9	46,7	7,0	-8,4	0,181
Prosent vedlikehold	21	58,0	23,9	18	70,2	18,4	-12,2	0,085
Prosent funksjonell vedlikehold	21	55,8	21,6	18	59,1	18,8	-3,3	0,620

Table 6-38: Maintenance vs. use of system development tools

From Table 6-38 we do not find any significant difference in maintenance between organizations that use system development tools and the organizations that do not use system development tools. We do not reject H23.

H24: There is no difference in the amount of maintenance-work between organizations that use system maintenance tools, and the organizations that do not use this.

	Bruker CASE			Bruker ikke CASE				
	N	Mean	SD	N	Mean	SD	Δ	p
Total andel vedlikehold	15	37,8	16,8	25	36,6	14,0	1,2	0,815
Vedlikehold hovedsakelig	13	40,1	16,9	13	42,3	13,5	-2,2	0,713
Prosent vedlikehold	15	61,4	23,2	25	66,5	22,5	-5,1	0,498
Prosent funksjonell vedlikehold	15	59,8	22,8	25	57,5	20,3	2,3	0,744

Table 6-39: Maintenance vs. use of system maintenance tools

In Table 6-39 we do not see any significant differences in maintenance between organizations that use system maintenance tools and organizations that do not use system maintenance tools. We do therefore not reject H24.

In the table below we will compare variables from [Holgeid 99] with variables from our investigation.

	2003 – 1 1998 - 2	N	Mean	SD	Δ	p
Total andel vedlikehold	1 2	52 52	35,9 41,4	15,6 20,6	-5,5	0,126
Feilretting	1 2	52 52	8,7 12,7	6,1 12,5	-4,0	**0,038**
Adaptiv	1 2	52 52	7,2 8,2	4,9 7,7	-1,0	0,454
Funksjonell	1 2	52 52	12,5 15,2	11,3 18,1	-2,7	0,365
Ikke-funksjonell	1 2	52 52	7,5 5,4	6,0 5,6	2,2	0,060
Total andel utvikling	1 2	52 52	21,9 17,1	17,8 17,1	4,8	0,167
Erstatning	1 2	52 52	9,7 7,7	8,4 10,1	2,1	0,259
Nyutvikling	1 2	52 52	12,2 9,5	13,8 13,3	2,7	0,312
Drift	1 2	52 52	23,1 23,0	16,4 14,4	0,1	0,982
Brukerstøtte	1 2	52 52	16,8 18,6	13,2 12,9	-1,8	0,478
Annet	1 2	52 52	2,3 0,0	7,4 0,0	2,3	**0,026**
Prosent vedlikehold	1 2	52 52	65,9 72,9	21,4 23,9	-7,0	0,118
Prosent utvikling	1 2	52 52	34,1 27,1	21,4 23,9	7,0	0,118
Prosent funksjonell utvikling	1 2	52 52	38,9 37,7	20,2 23,4	1,2	0,776
Prosent funksjonell vedlikehold	1 2	52 52	61,1 62,3	20,2 23,4	-1,2	0,776
Vedlikehold hovedsakelig	1 2	32 29	41,3 50,2	14,6 23,0	-8,9	0,074
Utvikling hovedsakelig	1 2	32 29	30,0 24,3	17,0 18,7	5,7	0,214

Table 6-40: comparison between variables from 1998 and 2003

We can see that there is a significant difference in “Feilretting” and “Annet”. There is significantly less “Feilretting” in our investigation than reported in [Holgeid 99].
We will look away from the difference in “Annet” here, because the investigation from 1998 did not have any data on this point. From Table 6-40, we see that the rest of our variables are not significantly different from the ones in 1998.

7 Discussion

In this chapter we will look at the results of the hypotheses in context. In the previous chapter we discussed the results from each hypothesis in detail. We will in this chapter look at the results on a higher level, and try to look at different ways to explain the results. We will present the explanations which we find most likely based on both the analyses that we have done, and earlier investigations.

7.1 Share maintenance

Most of the earlier investigations have reported that maintenance-work, compared to development, makes out between 40% and 60% ([Riggs 69], [Lientz and Swanson 80], [Nosek and Palvia 90], [Krogstie 94]). [Holgeid 99] reported 73% maintenance – when we look at development and maintenance-work alone. Compared to our investigation (66%), we see a decrease of 7%. 66% maintenance-work is still high compared to other investigations. The large percent maintenance-work in [Holgeid 99] was thought to drop back to “normal” after the Y2K problem was over, but it does not seem to do this.

We will list some possible reasons for the high percentage of maintenance-work, and discuss them closer below.

- Increased lifetime among the IT-systems leads to larger and unstructured systems due to constant maintenance.
- IT-systems are more difficult to maintain because of the increased user-demands and new technologies. New technology brings new aspects regarding maintenance.
- Use of consultants in the development phase, and little or no transfer of expertise to the maintenance-workers.
- Maintenance of old systems preferred instead of developing replacement-systems
- The organizations are more aware of maintenance-costs which lead to more preventive maintenance to save money in the longer-term.
- Less effective maintenance in the organizations, due to little use of methodology in development- and maintenance-work, combined with few organizing-controls during the maintenance-process.

One of the conclusions in [Henne 92] is that the low share traditional maintenance is due to the fact that the IT-systems were phased out very early. When we in our investigation, [Holgeid 99] and in [Krogstie 94] see that the proportion between share maintenance and development has changed from the result in [Henne 92], we can speculate if this is because of increased lifetime of the IT-systems – which leads to larger and unstructured systems due to maintenance through out the years. The average age of the main-systems in our investigation is lower than in [Holgeid 99] and [Krogstie 94], so the high maintenance share can therefore hardly be explained by this model.

One can imagine that the share maintenance has increased the last years because the IT-systems have become more difficult to maintain due to increased user-demands and the entry of new technology. The increase of IT in general and especially in the organizations has made that many organizations are dependent of IT. The high demands and new functional-areas which must be managed are likely to bring more complexity to the solutions that have to be maintained. Earlier the focus was on technology, but now the focus is more on satisfying the users. Focus on the users is crucial to the organizations, and is often the reason for expanding the IT-systems which makes them more difficult to maintain.

In our investigation we found that there is little use of IT-consultants among our respondents, as mentioned earlier this can be a result of the size of the respondent organizations, where smaller organizations are thought to use fewer IT-consultants than the larger organizations. We have not looked at what type of work the consultants do, but one explanation can be that smaller organizations hire IT-consultants to help them with the development only. This can make the maintenance of the IT-systems more difficult, due to lack of expertise-transfer to the employees who perform the maintenance. We have found in our investigation that there is no significant difference in share maintenance between organizations with few hired IT-consultants and organizations with many hired IT-consultants. There was also no significant difference in share maintenance between organizations where maintenance was “Alltid/ofte” performed by the same people who developed the system, and organizations where maintenance was “Av og til /skjelden/aldri” performed by the same people who developed the system. The high maintenance share can therefore hardly be explained by this model.

Many organizations have invested large amounts of money in today’s IT-systems. One would therefore intuitively think that maintenance of the old systems would be preferred in preference to developing replacement systems. We have however found out that 60% of the systems developed are replacement-systems, and the work with these systems is counted as functional maintenance, and not as traditional maintenance. The respondents report that the most important reason for the replacement-systems are “integrering med andre systemer”, “standardisering”, “ny teknisk arkitektur” and “det opprinnelige systemet var vanskelig å vedlikeholde”.

One other explanation for the increased share maintenance can be that the organizations are more aware of the maintenance-costs. Focus on maintenance-costs can lead to a increased preventive maintenance to save money in the longer-term. Further investigations have to be performed to reveal if increased preventive maintenance is the reason to the large share maintenance.

One last explanation can be that the organizations in general have become less effective to maintain the IT-systems due to little use of methodology in development- and maintenance-work, combined with few organizing-controls during the maintenance-process. We will look closer at this possible explanation in the next chapter which deals with traditional maintenance particularly.

7.2 Traditional maintenance

In this chapter we will discuss factor that looks to have influence on the share traditional maintenance in the organizations. With share traditional maintenance, we will here mean the share traditional maintenance in context with both the IT-department total work-time and in share development and traditional maintenance when we look away from other work.

Organizations with many main-systems, many different system-configurations and many different programming-languages seem to have less traditional maintenance than the rest of the organizations. This may look strange, but some of the explanation can be that organizations with a complex portfolio use organizing-controls and methodologies to control the alteration connected with maintenance. It is mostly the larger organizations that have the most complex portfolios – if the complexity can be decided from the number of main-systems, number of different system-configurations and number of different programming-languages used. It is as mentioned above the large organizations that have introduced the largest width of organizing-controls.

We can imagine that organizations with complex IT-systems are reluctant to make any changes because of the large risk of introducing new errors, and because large expenses are connected with making changes to older, over-complex systems. They may instead make small applications which are extensions to the old systems which no one wishes to maintain. These additional applications can be described as functional- extensions with an overlap towards existing program-code, and the work with these applications will therefore border on functional maintenance. This type of work will reduce the share traditional maintenance. Further investigations have to be performed to reveal if this type of factors have reduced the share traditional maintenance in the larger organizations.

7.3 Traditional maintenance in organizations with IT-departments which mostly work with development and maintenance

In this chapter we will look at different factors connected to traditional maintenance in organizations with IT-departments that mainly work with development and maintenance of IT-systems.

We mentioned as one possible explanation in chapter 7.1, that organizations may be more aware of the maintenance-costs, and will therefore focus more on preventive maintenance to save money in the longer-term. One would think that IT-departments that work primary with development and maintenance would focus more on improving the efficiency of this work than IT-departments that work primary with other things like network or user-support. This is however not supported in our result; we found that there is no significant difference in share maintenance between organizations with IT-departments that work primarily with development and maintenance and the organizations where the IT-departments work primarily with other things. This means that organizations where IT-departments work primarily with development and maintenance do not have a larger share maintenance than organizations where the IT-department work primarily with other things than development and maintenance.

7.4 Functional maintenance and functional development

In this chapter we will discuss factors that influence the share functional maintenance.

Henne found that share traditional maintenance was only 26% [Henne 92]. Henne discussed this sensational low share maintenance, and reported that a possible explanation for this was that the lack of procedures and maintenance-controls led to the decline of the systems, and that they were phased out early. The share traditional maintenance was low, and there were more replacement-systems. This is an important observation in context with functional maintenance and functional development, since it illustrates that the share traditional maintenance can not alone explain to which degree the organizations are succeeding with efficient maintenance. The purpose of maintenance can be said to be to maintain the organizations need for support from the IT-systems [Krogstie 95].

In our investigation there was only one hypothesis that was rejected for functional maintenance. Hypothesis 22 which looks at the difference in the amount of maintenance-work between organizations that use methodologies in the IT-systems lifecycles, and the organizations that do not use this. Where organizations that do not use methodologies in the IT-systems lifecycles have significantly more functional maintenance compared to the organizations that do use this.

8 Evaluation of the investigation

In this chapter we will evaluate the investigation by looking at different limitations related to the investigation.

8.1 Hypothesis-testing/correlation analysis

When testing a hypothesis we find out if we have to reject the hypothesis or not with a given significance-level. The correlation analysis shows us if there is a covariance between variables, and how the covariance is. Neither the hypothesis-test nor the correlation analysis tells us the real reason why a hypothesis is rejected or why the variables correlate. The hypothesis-results and the covariance may be influenced by other, untested, variables. We have in this investigation tried to find possible causes for the presented results from hypothesis-testing and correlation analysis, but the real cause is hard to prove.

The chosen significance-level is also decisive when it comes to rejecting or not rejecting a hypothesis. A low significance-level we give us fewer significant results than a high significance-level.

8.2 The respondents

The respondents in our investigation are mostly leaders (see chapter 5.1). The answers provided to us could be different if the respondents were mostly system-developers.

8.3 Generalization

We received approximately the same amount of respondents as [Holgeid 99], but our response rate was higher because of the smaller sample. We took a closer look at this in chapter 3, where we concluded that our response-rate was at the same level or higher than the investigations we compare with. A higher response-rate would have allowed us to generalize more.

The organizations that received our questionnaire were selected from the member-register of The Norwegian Computer Society. We have not investigated if this has had any influence on the results, but this is a potential uncertainty factor, since the organizations within a special grouping may be categorized by joint attributes which separate from the remaining organizations.

8.4 Quantitative/ qualitative

As mentioned in chapter 3, the investigation is mainly based on the analysis of the questionnaire.

The questionnaire is an effective method to collect answers from a large group of people, but it is not clear whether the respondents understand the questions equally. To perform interviews is a more secure method to ensure that the questions are understood equally, but are much more time-consuming. Using qualitative inquiry it is possible to look at conditions which often will not result from quantitative inquiries.

Because of the time-limit, we have not hade the time to perform interviews, and therefore it is difficult to ascertain to which degree we have succeeded in getting the respondents to understand the questionnaires equally. When comparing our results to other investigations, similar problems may occur, because the investigations are performed in different organizations, different geographical areas and different cultures.

8.5 Number of tests

We have in this investigation performed test on 24 hypotheses. It is worth mentioning that many tests can influence the results. For example if the number of significant results is 10, the results will be stronger if the number of performed test is 50 instead of 200. The number of tests performed in our investigation does not differ much from the number in investigations we compare us with. Other investigations in maintenance within IT-systems have tested far more hypotheses (for example [Martiniussen 96]). Many tests on the same data-foundation can be a source of error where significant results can arise without necessarily presenting direct and explainable connections.

9 Conclusion and further work
In this chapter we will sum up the main-results from the investigation, and propose further work on empirical investigations within maintenance of IT-systems.

9.1 Conclusion
In this chapter we will sum up the main-results from the performed investigation.

9.1.1 Work allocation
The investigations by [Henne 92], [Krogstie 94] and [Holgeid 99] have respectively reported maintenance makes out 26%, 59% and 73% of the work. In our investigation we have found out that maintenance makes out 65,9% - when we look at development and maintenance only.

When we look at functional development in proportion to functional maintenance, functional development makes out 38,9%. Functional maintenance makes up the remaining 61,1%. This is similar to the results in [Holgeid 99] where functional development makes out 37,7% and functional maintenance the remaining 62,3%, but significantly different from what was reported in [Krogstie 94].

9.1.2 Size and type of organization
We did not find any significant differences in the maintenance-work performed vs. number of employees in the organizations. [Holgeid 99] reported that organization with many employees performed less traditional maintenance. He reported use of methodologies as one explanation. One could assume that we would find similar differences, but this was not the case.

9.1.3 Importance of IT
There were no significant differences in maintenance-work performed vs. the importance of IT in the organizations. This means that organizations who report that IT is of "Absolutt strategisk betydning" do not perform more maintenance than organizations who report IT to be of less importance. It is here important to point out that almost 80% of the respondents reported the importance of IT to be of "Absolutt strategisk betydning".

9.1.4 Internal qualifications
Qualifications can be measured in different ways, we have in this investigation decided to look at work-experience and education. We did not find any significant difference in maintenance between organizations where system-developers had little experience, and organizations with system-developers with more experience.

9.1.5 Complexity of the portfolio
The complexity of the portfolio has been measured by different characteristics like number of main-systems, dependency between IT-systems, number of system-configurations, number of programming-languages in use and number of data-bases.
We found significant difference in "Vedlikehold hovedsakelig" between organizations with few main-systems and organizations with many main-systems. Organizations with

many main-systems had significantly more “Vedlikehold hovedsakelig” than organizations with few main-systems. There was also significant less “Prosent vedlikehold” in organizations with many different system-configurations compared to organizations with few system-configurations. Organizations with many different programming-languages had significantly less “Total andel vedlikehold”, “Vedlikehold hovedsakelig” and “Prosent vedlikehold” than organizations that used few programming-languages.

9.1.6 Organizing

[Lientz and Swanson 80] reported significantly less maintenance-work in organizations where development- and maintenance-work was organized differently. [Holgeid 99] did not find any significant difference between these organizations. We did not either find any significant differences here.

9.1.7 Use of methods

We found that organizations that use of methodologies in the IT-systems lifecycle had significantly less “Prosent funksjonell vedlikehold” than organizations that do not use methodologies.

We took a closer look at the use of methodologies in some different phases of the IT-systems lifecycles and found that organizations that use methodologies in the analysis-phase had significantly less “Total andel vedlikehold”, “Vedlikehold hovedsakelig” and “Prosent vedlikehold” than organizations that do not use methodologies in this phase. Organizations that use methodologies in the requirement-specification phase have significantly less “Prosent vedlikehold” than organizations that do not use this. There was also a significant difference between organizations that use methodologies in the implementation-phase and those that do not, where organizations that use methodologies in this IT-system lifecycle phase had significantly less “Prosent funksjonell vedlikehold” than organizations that do not use methodologies in this phase of the IT-systems lifecycle.

9.1.8 Use of tools

Here we did neither find any significant differences between organizations that use system development tools and organizations that do not nor between organizations that use system maintenance tools and the ones that do not.

9.2 Further work

As in most investigations we did also in our find new areas which could be interesting to examine further. We will here present some of the areas which may create some basis for new investigations.

As mentioned in chapter 3, the data-basis for our investigation was gathered by questionnaires, and can be reckoned as quantitative. It could be interesting to make a similar investigation based on qualitative data to compare the results.

As in the investigation from [Holgeid 99] we found that the complexity of the IT-systems portfolio mostly correlated in a negative way with share maintenance. Intuitively one would thing opposite. We have looked at some explanations for this, but further investigations can look at different portfolio characteristics and in what way they influence maintenance-types and extent.

It could also be interesting to look closer at the difference between organizations that use pre-defined methods and development/maintenance tools in their work and the ones that do not, and how the use of this affects maintenance-work.

It would also be interesting to make a follow-up investigation to our investigation. In both [Krogstie 94], [Holgeid 99] and our own investigation the respondents were organizations that are members of The Norwegian Computer Society. The respondents from these organizations were mostly leaders, and it is difficult to know how this has affected the results. In a follow-up investigation, one could choose the respondents based on a different criterion to compare with our results.

References

[Ask 94] F. F. Ask:
“Elementær statistikk, en pedagogisk innføring”, 2. utgave, UNIKURS 1994.

[Bergersen 90] L. Bergersen:
“Prosjektadministrasjon i systemutvikling. Aktiviteter i planlegningsfasen som påvirker suksess”, PhD avhandling, ORAL, NTH, Trondheim, Norge 1990

[Boehm 76] Barry W. Boehm:
”Software engineering”, IEEE Transactions on Computing, December 1976

[Boehm 88] Barry W. Boehm:
“A Spiral Model of Software Development and Enhancement”, IEEE Computer, Vol. 21, Nr 5, May 1988.

[Bronstein and Okamoto 81] G. M. Bronstein and R. I. Okamoto:
“I’m OK, You’re OK, Maintenance is OK.”, Computerworld, 1985, 19, page 45

[Bush 88] E. Bush:
”A CASE for existing systems”, Language Technology White Paper, page 4, 1988

[Couger 85] J. D. Couger:
“Motivating Maintenance Personnel”, Computerworld, 12 August 1985.

[Dekleva 92] S. M. Dekleva:
“The influence of the information systems development approach on maintenance”, MIS Quarterly, side 355 – 372, September 1992.

[Hale et al 91] David Hale, Dwight A. Haworth, Shane Sharpe:
“Characteristics of empirical Software Maintenance Studies: 1980 – 1989”, Software Maintenance: Research And Practice, Vol. 3, pages 1-15, 1991.

[Henne 92] A. Henne:
“Information systems maintenance – problems or opportunities?”, i *Proceedings of Norsk Informatikk Konferanse 1992* (NIK’92), pages 91-104, Tromsø, Norge, 1992.

Holgeid, K.K., J. Krogstie, and D.I.K. Sjøberg, A study of development and maintenance in Norway: Assessing the efficiency of information systems support using functional maintenance. Information and Software Technology, 2000. 42: p. 687-700.

[Holgeid 99] K. K. Holgeid:
“Utvikling og vedlikehold av IT-systemer i norske bedrifter”, M.Sc. thesis, Institute Of Informatics of Oslo, April 1999.

[House 93] C. House:
“Case studies in the CASE industry”, Keynote Speech, CASE ‘93

[Jørgensen 94] Magne Jørgensen:
”Empirical Studies of Software Maintenance”, Research Report 188 ISBN 82-7368-098-3

[Krogstie 94] John Krogstie:
”Survey Investigation: Development and Maintenance of Information Systems”, IDT, NTH, 1994, pages 1-2

[Krogstie and Sølvberg 94] John Krogstie and Arne Sølvberg:
“Software maintenance in Norway: A survey investigation”, in Proceedings of International Conference of Software Maintenance, ICSM ‘94, IEEE, September 1994, pages 304-313

[Krogstie 95] John Krogstie:
“On the Distinction between Functional Development and Functional Maintenance”, i Journal of Software Maintenance, IEEE, pages 383-403, 1995

[Lehman and Belady 85] M. M. Lehman and L. Belady:
“Program Evolution, processes of Software Change. A.P.I.C Studies in Data Processing No. 27. Academic Press, London.

[Lientz and Swanson 80] B. P. Lientz and E. B. Swanson:
”Software Maintenance Management”, Addison-Wesley, Reading, MA. 1980, ISBN 0-201-04205-3

[Liu 76] C. Liu:
”A look at software maintenance”, Datamation, 22(11):51-55, November 1976

[Lyons 81] M. J. Lyons:
”Salvaging your software asset (tool based maintenance)”, i AFIPS Conference Proceedings on 1981 National Computer Conference, pages 337-342, Chicago, USA, 4-7 May 1981

[Moad 90] J. Moad:
”Maintaining the competitive edge”, Datamation, pages 61-66, February, 1990

[Norusis 92] M. J. Norusis:
“SPSS for Windows: SPSS Base Application Guide”, SPSS Inc., Chicago, Illinois, USA, 1992

[Nosek and Palvia 90] J. T. Nosek and P. Palvia:
“Software maintenance management: Changes in the last decade”, Journal of Software Maintenance, 2:157-174, 1990.

[Ogden 72] J. L. Ogden:
”Designing reliable software” Datamation, 21:71-78, July 1972

[Reutter 81] J. Reutter:
”Maintenance is a management problem and a programmer’s opportunity”, I AFIPS Conference Proceedings on 1981 National Computer Conference, pages 343-347, Chicago, USA, 4-7 May 1981

[Reynolds 77] C. H. Reinolds:
“Issues in Centralization.”, Datamation, 1977, 23, pages 91 - 94

[Riggs 69] R. Riggs:
”Computer systems maintenance”, Datamation, 15:227-235, November 1969

[Scott and Farley 88] T. Scott and D. Farley:
”Slashing software maintenance costs”, Business Software Review, March 1988

[Swanson 76] E. B. Swanson:
”The dimensions of maintenance”, in proceedings of the Second International Conference on Software Engineering, side 492-497, San Francisco, USA, 13-15 august, 1976

[Swanson and Beath 89] E. B. Swanson and C. M. Beath:
“Maintaining Information Systems in Organizations”, Wiley Series in Information Systems, John Wiley & Sons, 1989.

[Vessey and Weber 83] I. Vessey and R. Weber:
“Some factors affecting programming repair maintenance”, Communications of the ACM, 26(2):128-134, February 1983

[Swede and Vliet 94] V. van Swede and H. van Vliet:
“Consistent development: Results of a First Empirical Study of the Relation between Project Scenario and Success”, i “Proceedings of the 6th International Conference on Advanced Information Systems Engineering (CAiSE’94), nr 811 in Lecture Notes in Computer Science, Springer Verlag, June 1994

[Wilson 71] T. P. Wilson:
“Critique of ordinal variables”, in H. M. Balalock junior, editor, Causal Models in the Social Sciences, Aldine Atherton, 1971

[Wonnacott et al 90] T. H. Wonnacott and R. J. Wonnacott:
“Introductory statistics”, 5. edition, John Wiley & Sons, 1990

Appendix

Appendix 1: Letter of information

Datainnsamling
Utvikling og vedlikehold av informasjonssystemer

Til IT-ansvarlig

Den norske Dataforenings faggruppe for metoder, Simula Research Laboratory, SINTEF og Universitetet i Oslo gjennomfører en undersøkelse blant norske bedrifter med hensyn på nåværende praksis innen utvikling og vedlikehold av IT-systemer .

Ca. 200 bedriftsmedlemmer i Den norske Dataforening får i denne forbindelse tilsendt dette spørreskjemaet. Svarene vil gi grunnlag for omfattende analyser og videre resultatpublisering.

Alle opplysninger vil bli behandlet strengt konfidensielt. Det vil dermed ikke bli publisert data som kan "spores tilbake" til en spesifikk bedrift eller person.

Dere som svarer på undersøkelsen vil få tilgang på sluttresultatene fra undersøkelsen.
Dessuten vil du (eventuelt bedriften hvis ønskelig) motta 500 kr kontant skattefritt.

Spørreskjemaet fylles ut via web: http://194.143.23.38/sese innen........................ til:
Arthur Jahr
Hedmarksgata 7. Leil: 4005
0658 Oslo

Ta gjerne kontakt med Arthur Jahr på tlf: 91377324

Arthur Jahr,
institutt for informatikk, UiO

John Krogstie
SINTEF Tele og Data

Dag Sjøberg
Simula Research Laboratory

Appendix 2: The questionnaire

SPØRRESKJEMA, utvikling og vedlikehold av informasjonssystemer

Informasjons som fremkommer på denne siden vil ikke bli koblet med annen informasjon som måtte komme frem i de øvrige spørsmålene i undersøkelsen.

Bedriftens navn: ______________________________

Respondentens (ditt) navn: ____________________________________

Veiledning for utfyllere

Spørreskjemaet vil enklest kunne besvares av en IT-sjef eller en som innehar en tilsvarende stilling i bedriften. Svarene må være basert på de rutiner og den praksis som organisasjonen fører i dag. I tillegg til at en rutine eksisterer, må den som svarer vurdere effektiviteten og kvaliteten til rutinen.

Relevansen til noen av spørsmålene vil være avhengig av svarene på tidligere spørsmål. Det er viktig at alle relevante spørsmål blir besvart. Hvis enkelte spørsmål ikke er relevante, la feltene være blanke.

Respondentens stilling: ____Leder
____Prosjektleder
____Systemutvikler

Ansettelsesforhold: ____Fast
____Midlertidig
____Innleid konsulent
____Annet Spesifiser: ______________________

Formell utdannelse:____

Antall års EDB erfaring: ____

Kort beskrivelse av type erfaring, arbeidsoppgaver, ansvar m.m. i nåværende jobb:
____
____
____

1. Type organisasjon? (sett ett kryss)

a. ____Tele og data
b. ____Bank og forsikring
c. ____Offentlig forvaltning
d. ____Helsevesen
e. ____Reise og transport
f. ____Handel
g. ____Industri
h. ____Tjenesteyting/konsulentvirksomhet
i. ____Annet Spesifiser:____

2. Er IT av strategisk betydning for bedriften?

Absolutt av strategisk betydning __5 __4 __3 __2 __1 Ikke av strategisk betydning
Kommentar:
____
____
____

3. Hvor mange ansatte har bedriften? ________________

4. Hva er det årlige budsjettet for dataavdelingen inklusive maskinvare, programvare og personell (oppgitt i millioner kroner og uten avskrivninger)?

		2003
a.	mer enn 50	____
b.	mellom 40 og 50	____
c.	mellom 30 og 40	____
d.	mellom 20 og 30	____
e.	mellom 10 og 20	____
f.	mellom 1 og 10	____
g.	mindre enn 1	____

5. På bakgrunn av de totale utførte timeverk i løpet av et år, hvor mye (i prosent) brukes til:

	%	
a.	___	Rette feil i systemer som er i produksjon
b.	___	Tilpasse systemer til endret teknisk arkitektur
c.	___	Utvikle ny funksjonalitet i eksisterende system
d.	___	Forbedre ikke-funksjonelle egenskaper (f.eks. ytelse) i eksisterende systemer
e.	___	Utvikle nye system som overlapper/erstatter gamle systemer funksjonelt sett
f.	___	Utvikle nye system for å dekke nye funksjonsområder
g.	___	Drift
h.	___	Brukerstøtte
i.	___	Annet Spesifiser:____
sum:	100%	

Svaret ovenfor er:
a. ___Rimelig nøyaktig, basert på gode data
b. ___Et grovt estimat, basert på minimale data
c. ___En best mulig gjetning, ikke basert på noen data

<u>Begrepet '**vedlikehold**' omfatter i resten av skjemaet oppgaver av type a, b, c og d i spørsmål 5.</u>

6. Hvor mange personer er ansatt i dataavdelingen (omregnet til fulltidsansatte)? ____personer

7. Hvor mange av disse er systemutviklere (omregnet til fulltidsansatte)? ____personer

8. Hva er fordelingen av systemutviklerne med hensyn til:

a. **hvor lenge de har arbeidet i avdelingen?**

0-1 år	_______personer
1-3 år	_______personer
3-6 år	_______personer
6-10 år	_______personer
Mer enn 10 år	_______personer

b. **total erfaring?**

0-1 år	_______personer
1-3 år	_______personer
3-6 år	_______personer
6-10 år	_______personer
Mer enn 10 år	_______personer

Svaret ovenfor er:
a. ___Rimelig nøyaktig, basert på gode data
b. ___Et grovt estimat, basert på minimale data
c. ___En best mulig gjetning, ikke basert på noen data

9. Hvilke utdanningsbakgrunn har systemutviklerne?

Siv.ing, Cand.scient eller tilsvarende med spesialisering innen datateknikk	______personer
Som over, men annen spesialisering (inkluderer også siv.øk. etc.)	______personer
Ingeniørhøyskole, DH-kandidat eller annet med spesialisering innen datateknikk	______personer
Som over, men med annen spesialisering	______personer
Uten høyere utdannelse	______personer

10. Hvor mange innleide konsulenter innen systemutvikling har avdelingen i gjennomsnitt over et år (omregnet til fulltidsansatte)? _______personer

11. Er de som arbeider med vedlikehold organisert forskjellige fra de som driver nyutvikling?

a. ja, formelt ____
b. ja, uformelt ____
c. nei ____

Kommentar:
____
____
____

12. Blir vedlikehold av informasjonssystemer utført av de som opprinnelig laget systemet?

Alltid __5 __4 __3 __2 __1 Aldri __Vet ikke

Kommentar:
____
____
____

13. Blir vedlikehold av informasjonssystemer utført som en opplæringsaktivitet i organisasjonen?

Alltid __5 __4 __3 __2 __1 Aldri __Vet ikke

Kommentar:
____
____
____

14. Hvor mange større systemer (hovedsystemer) er i produksjon i organisasjonen?
_______systemer

____

15. Hvilke områder dekker disse hovedsystemene (så som lønn, lagerstyring, regnskap osv.)?

16. Hvor mange sluttbrukere bruker disse systemene?

a. Internt i bedriften:_______personer

Svaret ovenfor er:
a. ___Rimelig nøyaktig, basert på gode data
b. ___Et grovt estimat, basert på minimale data
c. ___En best mulig gjetning, ikke basert på noen data

b. Eksternt: _______personer

Svaret ovenfor er:
a. ___Rimelig nøyaktig, basert på gode data
b. ___Et grovt estimat, basert på minimale data
c. ___En best mulig gjetning, ikke basert på noen data

17. Hva er aldersfordelingen til eksisterende hovedsystemer (regnet i år etter første installasjon)?

0-1 år	_______systemer
1-3 år	_______systemer
3-6 år	_______systemer
6-10 år	_______systemer
Mer enn 10 år	_______systemer

18. Hvordan er de forskjellige hovedsystemene utviklet?

Utviklet av dataavdelingen	_______systemer
Utviklet i brukeravdelingen i bedriften	_______systemer
Utviklet av et utenforstående selskap	_______systemer
Pakkeløsning, med store interne tilpasninger	_______systemer
Pakkeløsning, med små interne tilpasninger	_______systemer
Sammensatt av komponenter av delvis ukjent opphav (f.eks. ved gjenbruk av komponenter man henter ned fra Internet)	_______systemer

19. Hvor mange av hovedsystemene er avhengig av data fra andre systemer?
_______systemer

Svaret ovenfor er:
a. ___Rimelig nøyaktig, basert på gode data
b. ___Et grovt estimat, basert på minimale data
c. ___En best mulig gjetning, ikke basert på noen data

20. **Hvilke maskinvare og systemprogramvare er i bruk, og hvor mange systemer understøttes av konfigurasjonen? (Stormaskin, Windows PC/Server, Unix er eksempler på konfigurasjoner).**

Konfigurasjon	Antall systemer
____	_______
____	_______
____	_______
____	_______
____	_______

21. **Hvilke programmeringsspråk er i bruk?**

Språk	Antall systemer	
COBOL	_______	
Assembler	_______	
C	_______	
C++	_______	
Java	_______	
4 GL språk	_______	Spesifiser:____
Andre	_______	Spesifiser:____

22. **Hvilke typer databasesystemer er i bruk?**

	Antall databaser	
Hierarkiske databaser	_______	
Nettverksdatabaser	_______	
Relasjonsdatabaser	_______	
Objektorienterte databaser	_______	
Annet	_______	Spesifiser:________________________________

23. **Hvor mange nye systemer er for tiden under utvikling?** _______systemer

24. **Av totalt antall nye systemer, hvor mange av disse er erstatningssystemer, det vil si hvor mange av de nye systemene dekker hovedsakelig funksjonalitet som alt er dekket i eksisterende systemer?** _______systemer

25. **Hva er aldersfordelingen på de systemene som eventuelt erstattes?**

0-1 år	_______systemer
1-3 år	_______systemer
3-6 år	_______systemer
6-10 år	_______systemer
Mer enn 10 år	_______systemer

26. **Ved utvikling av erstatningssystem, hva er de viktigste grunnene for at de blir erstattet (gi score fra 5-1 på alle punktene nedenfor)?**

		Svært viktig 5 4 3 2 1 Uviktig
a.	Svært vanskelig å vedlikeholde eksisterende system	__ __ __ __ __
b.	Svært vanskelig å drifte eksisterende system	__ __ __ __ __
c.	Svært vanskelig å bruke eksisterende system	__ __ __ __ __
d.	Finnes alternativ pakkeløsning	__ __ __ __ __
e.	Finnes alternativ applikasjonsgenerator	__ __ __ __ __
f.	Ny teknisk arkitektur i organisasjonen	__ __ __ __ __
g.	Standardisering med resten av organisasjonen	__ __ __ __ __
h.	Integrering med andre nye eller eksisterende systemer	__ __ __ __ __
i.	Annet, spesifiser:____	__ __ __ __ __

Karakterforklaring:
5: Faktoren er svært viktig
4: Faktoren er viktig
3: Faktoren er noe viktig
2: Faktoren er lite viktig
1: Faktoren er uviktig

27. **Ved utvikling av erstatningssystemer og nye systemer med overlappende funksjonalitet til eksisterende system, i hvor stor grad er man i stand til å gjenbruke spesifikasjoner og design?**

Svært mye __5 __4 __3 __2 __1 Svært lite

Kommentar:
____
____
____

28. **Ved utvikling av erstatningssystemer og nye systemer med overlappende funksjonalitet til eksisterende system, i hvor stor grad er man i stand til å gjenbruke kode?**

Svært mye __5 __4 __3 __2 __1 Svært lite

Kommentar:
____
____
____

29. Hvilke organisatoriske kontroller er etablert for vedlikehold av informasjonssystemer?

a. ___ Alle brukerkrav som kommer inn blir dokumentert
b. ___ Endringsforslag blir klassifisert etter type og viktighet
c. ___ Alle endringsforslag gjennomgår en konsekvensanalyse og kostnadsestimering
d. ___ Alle endringer av programvaren blir dokumentert
e. ___ Alle endringer av informasjonssystemet blir testet før systemet settes i produksjon
f. ___ Med unntak av driftstruende feil blir alle endringer samlet opp for periodisk implementasjon
g. ___ Ved akseptansetest av endringer, sjekkes også at den tilliggende dokumentasjon er oppdatert
h. ___ Brukere som etterspør endringer får beskjed både hvis endringsforslaget gjennomføres eller underkjennes
i. ___ Man bruker samme rutiner for endringsforslag som kommer fra dataavdelingen som for endringsforslag som kommer fra brukergrupper
j. ___ Det gjennomføres en formell gjennomgang av systemet periodisk
k. ___ Økonomiske utstyrkostnader som er forbundet med drift og vedlikehold av informasjonssystemet belastes brukergruppene
l. ___ Personellkostnader forbundet med drift og vedlikehold av informasjonssystemet belastes brukergruppene

Kommentar:
___
___
___

30. I hvilken grad benyttes dokumentasjonen i vedlikeholdsarbeidet (kryss av på alle punktene nedenfor)?

		Ja	Nei	Ikke tilgjengelig
a.	Systemdokumentasjon	__	__	__
b.	Testdokumentasjon	__	__	__
c.	Brukerdokumentasjon	__	__	__
d.	Programmeringsspråksmanualer	__	__	__

Kommentar:
___
___
___

31. I hvilken grad er dokumentasjonen oppdatert/samstemt med det faktiske system (gi score fra 5-1 på alle punktene nedenfor)?

		I stor grad	5	4	3	2	1	I liten grad
a.	Systemdokumentasjon		__	__	__	__	__	
b.	Testdokumentasjon		__	__	__	__	__	
c.	Brukerdokumentasjon		__	__	__	__	__	

Kommentar:
___
___
___
___

32. I hvilke deler av livssyklusen til IT-systemene anvendes en på forhånd definert metode (sett ett eller flere kryss)?

			Hvis ja, hvilke(n) metode(r) brukes
a.	___	Planlegging	___
b.	___	Analyse	___
c.	___	Kravspesifikasjon	___
d.	___	Design	___
e.	___	Implementasjon	___
f.	___	Testing	___
g.	___	Utrulling	___
h.	___	Drift	___
i.	___	Vedlikehold	___
j.	___	Prosjektledelse	___

Kommentar:
___
___
___

Begrepet ”systemutviklingsverktøy” omfatter nedenfor alle former for automatiserte verktøy for utvikling/vedlikehold av informasjonssystemer. Et eksempel på et slikt verktøy er et modelleringsverktøy for analyse og design slik som ”Rational Rose”.

33. Anvendes systemutviklingsverktøy ved vedlikehold av informasjonssystemer?

Ja:__ Nei:__ Hvis ja – produktnavn:___

34. Anvendes systemutviklingsverktøy ved nyutvikling av informasjonssystemer?

Ja:__ Nei:__ Hvis ja – produktnavn:___

35. Hvilke av følgende deler av livssyklusen dekkes i dag gjennom anvendelse av systemutviklingsverktøy?

a. ___ Planlegging
b. ___ Analyse
c. ___ Kravspesifikasjon
d. ___ Design
e. ___ Implementasjon
f. ___ Testing
g. ___ Utrulling
h. ___ Drift
i. ___ Vedlikehold
j. ___ Prosjektledelse

Kommentar:
___
___
___

___

36. Hvor lenge har man brukt disse verktøyene i organisasjonen? _______ år

Kommentar:
____
____
____

37. Hvor mange av de eksisterende hovedsystemene (oppgitt under spørsmål 14) understøttes av systemutviklingsverktøyene? _______systemer

Svaret ovenfor er:
a. ___Rimelig nøyaktig, basert på gode data
b. ___Et grovt estimat, basert på minimale data
c. ___En best mulig gjetning, ikke basert på noen data

Kommentar:
____
____
____

38. I hvilken grad er følgende utsagn et problem ved vedlikehold av informasjonssystemer slik du bedømmer det (gi score fra 5-1 på alle punktene nedenfor)?

		Alvorlig problem 5 4 3 2 1 Intet problem
a.	Utskifting av personell	__ __ __ __ __
b.	Kvaliteten av systemdokumentasjonen	__ __ __ __ __
c.	Endringer av maskinvare og systemprogramvare	__ __ __ __ __
d.	Brukerkrav for utvidelser og forbedringer	__ __ __ __ __
e.	Ferdigheter til vedlikeholdspersonell	__ __ __ __ __
f.	Kvaliteten til det originale programmet	__ __ __ __ __
g.	Tilgjengelighet på vedlikeholdspersonell	__ __ __ __ __
h.	Konkurrerende behov om vedlikeholdspersonell	__ __ __ __ __
i.	Manglende interesse fra brukere	__ __ __ __ __
j.	Systemet feiler under operativ drift	__ __ __ __ __
k.	Manglende brukerforståelse av systemet	__ __ __ __ __
l.	Datalagringskrav	__ __ __ __ __
m.	Maskinhastighet	__ __ __ __ __
n.	Motivasjonen til vedlikeholdspersonell	__ __ __ __ __
o.	Vedlikeholdspersonellets produktivitet	__ __ __ __ __
p.	Pålitelighet til maskin og systemprogramvare	__ __ __ __ __
q.	Dataintegritet i applikasjonen	__ __ __ __ __
r.	Urealistiske brukerforventninger	__ __ __ __ __
s.	Programmeringsstandarder ikke brukt	__ __ __ __ __
t.	Trange budsjetter	__ __ __ __ __
u.	Mangelfull opplæring av brukere	__ __ __ __ __
v.	Utskiftninger i brukerorganisasjonen	__ __ __ __ __
w.	Ledelsen støtter ikke bruk av applikasjonen	__ __ __ __ __
x.	Annet, spesifiser:___________________________________	__ __ __ __ __

Karakterforklaring:
5: Faktoren ses på som et alvorlig problem
4: Faktoren er et større problem
3: Faktoren er et mindre problem
2: Faktoren er et lite problem
1: Faktoren er intet problem

Kommentar:

Vær så snill å se over skjemaet en gang til for å forvisse deg om at du ikke har hoppet over noen spørsmål. Tusen takk for innsatsen!

Appendix 3: Enclosed Letter from the Norwegian Computer Society

Oslo

Kartlegging av utvikling og vedlikehold av informasjonssystemer

Arthur Jahr, hovedfagsstudent ved Institutt for informatikk ved Universitetet i Oslo, ønsker å kartlegge og analysere utvikling og vedlikehold av informasjonssystemer i norske bedrifter. Resultatet av analysen vil bli gjort tilgjengelig for respondenten, og kan være til nytte i bedriftens egen IT-virksomhet.

Undersøkelsen følger opp tilsvarende undersøkelser foretatt i Dataforeningens regi i 1993 og 1998, og støttes i tillegg av Simula Research Laboratory og SINTEF.

Vedlikeholdsproblematikken har bare i liten grad vært gjenstand for akademiske studier. Dataforeningen ønsker å støtte akademiske arbeider som tar for seg aktuelle praktiske problemstillinger og oppfordrer alle utvalgte bedrifter til å besvare spørreskjemaet.

Med vennlig hilsen

John Krogstie
Den Norske Dataforening
Leder for Faggruppen for metoder og arkitektur

Appendix 4: Reminder

Purrebrev

Datainnsamling: Utvikling og vedlikehold av informasjonssystemer.
Jeg viser til tidligere utsendt spørreskjema, og tillater meg med dette å sende en påminnelse til de bedrifter som ikke har besvart skjemaet.

Det er avgjørende for undersøkelsen at flest mulig svarer. Utfyllingen av spørreskjemaet tar erfaringsmessig mellom 30 og 45 minutter. Hvis spørsmålene ikke er relevant for Dem, er det fint om skjemaet videreformidles til den IT-ansvarlige, eller en som innehar samme rolle i deres bedrift.

Til IT-ansvarlig

Den norske Dataforening, Simula Research Laboratory og Universitetet i Oslo gjennomfører en undersøkelse blant norske bedrifter med hensyn på nåværende praksis innen utvikling og vedlikehold av IT-systemer .

Ca. 200 bedriftsmedlemmer i Den norske Dataforening får i denne forbindelse tilsendt dette spørreskjemaet. Svarene vil gi grunnlag for omfattende analyser og videre resultatpublisering.

Alle opplysninger vil bli behandlet strengt konfidensielt. Det vil dermed ikke bli publisert data som kan "spores tilbake" til en spesifikk bedrift eller person.

Dere som svarer på undersøkelsen vil få tilgang på sluttresultatene fra undersøkelsen.
Dessuten vil du (eventuelt bedriften hvis ønskelig) motta 500 kr kontant skattefritt.

Vi håper Du kan hjelpe oss med dette.

Spørreskjema bes vennligst returnert så snart som mulig, og innen 15.januar 2004.
Det er å foretrekke at nettbaserte skjemaet besvares, men hvis det skulle være ønskelig å besvare skjemaet i papirform, kan denne sendes til:

Arthur Jahr
Hedmarksgata 7. Leil: 4005
0658 Oslo

Ved eventuelle problemer med for eksempel glemt brukernavn og/eller passord, vennligst kontakt Arthur Jahr på e-mail: arthurj@ifi.uio.no eller på tlf: 91377324.

På forhånd takk!